

课后答案网 您最真诚的朋友



www.hackshp.cn网团队竭诚为学生服务，免费提供各门课后答案，不用积分，甚至不用注册，旨在为广大学生提供自主学习的平台！

课后答案网：www.hackshp.cn

视频教程网：www.efanjy.com

PPT课件网：www.ppthouse.com

课后答案网
www.hackshp.cn

第 1 章 计算机系统结构的基本概念

1、有一个计算机系统可按功能分成 4 级，每级的指令互不相同，每一级的指令都比其下一级的指令在效能上强 M 倍，即第 i 级的一条指令能完成第 $i-1$ 级的 M 条指令的计算量。现若需第 i 级的 N 条指令解释第 $i+1$ 级的一条指令，而有一段第 1 级的程序需要运行 Ks ，问在第 2、3 和 4 级上一段等效程序各需要运行多长时间？

答：第 2 级上等效程序需运行： $(N/M)*Ks$ 。第 3 级上等效程序需运行： $(N/M)*(N/M)*Ks$ 。第 4 级上等效程序需运行： $(N/M)*(N/M)*(N/M)*Ks$ 。

2、硬件和软件在什么意义上是等效的？在什么意义上又是不等效的？试举例说明。

答：软件和硬件在逻辑功能上是等效的，原理上，软件的功能可用硬件或固件完成，硬件的功能也可用软件模拟完成。只是反映在速度、价格、实现的难易程度上这两者不同。

3、试以实例说明计算机系统结构、计算机组成与计算机实现之间的相互关系与影响。

答：计算机系统结构、计算机组成、计算机实现互不相同，但又相互影响。

(1) 计算机的系统结构相同，但可采用不同的组成。如 IBM370 系列有 115、125、135、158、168 等由低档到高档的多种型号机器。从汇编语言、机器语言程序设计者看到的概念性结构相同，均是由中央处理机/主存，通道、设备控制器，外设 4 级构成。其中，中央处理机都有相同的机器指令和汇编指令系统，只是指令的分析、执行在低档机上采用顺序进行，在高档机上采用重叠、流水或其它并行处理方式。

(2) 相同的组成可有多种不同的实现。如主存器件可用双极型的，也可用 MOS 型的；可用 VLSI 单片，也可用多片小规模集成电路组搭。

(3) 计算机的系统结构不同，会使采用的组成技术不同，反之组成也会影响结构。如为实现 $A:=B+CD:=E * F$ ，可采用面向寄存器的系统结构，也可采用面向主存的三地址寻址方式的系统结构。要提高运行速度，可让相加与相乘并行，为此这两种结构在组成上都要求设置独立的加法器和乘法器。但对面向寄存器的系统结构还要求寄存器能同时被访问，而对面向主存的三地址寻址方式的系统结构并

无此要求，倒是要求能同时形成多个访存操作数地址和能同时访存。又如微程序控制是组成影响结构的典型。通过改变控制存储器中的微程序，就可改变系统的机器指令，改变结构。如果没有组成技术的进步，结构的进展是不可能的。

综上所述，系统结构的设计必须结合应用考虑，为软件和算法的实现提供更多更好的支持，同时要考虑可能采用和准备采用的组成技术。应避免过多地或不合理地限制各种组成、实现技术的采用和发展，尽量做到既能方便地在低档机上用简单便宜的组成实现，又能在高档机上用复杂较贵的组成实现，这样，结构才有生命力；组成设计上面决定于结构，下面受限于实现技术。然而，它可与实现折衷权衡。例如，为达到速度要求，可用简单的组成但却是复杂的实现技术，也可用复杂的组成但却是一般速度的实现技术。前者要求高性能的器件，后者可能造成组成设计复杂化和更多地采用专用芯片。

组成和实现的权衡取决于性能价格比等因素；结构、组成和实现所包含的具体内容随不同时期及不同的计算机系统会有差异。软件的硬化和硬件的软件都反映了这一事实。VLSI 的发展更使结构组成和实现融为一体，难以分开。

4、什么是透明性概念？对计算机系统结构，下列哪些是透明的？哪些是不透明的？

存储器的模 m 交叉存取；浮点数据表示；I/O 系统是采用通道方式还是外围处理机方式；数据总线宽度；字符行运算指令；阵列运算部件；通道是采用结合型还是独立型；PDP-11 系列的单总线结构；访问方式保护；程序性中断；串行、重叠还是流水控制方式；堆栈指令；存储器最小编址单位；Cache 存储器。

答：透明指的是客观存在的事物或属性从某个角度看不到。

透明的有：存储器的模 m 交叉存取；数据总线宽度；阵列运算部件；通道是采用结合型还是独立型；PDP-11 系列的单总线结构；串行、重叠还是流水控制方式；Cache 存储器。

不透明的有：浮点数据表示；I/O 系统是采用通道方式还是外围处理机方式；字符行运算指令；访问方式保护；程序性中断；堆栈指令；存储器最小编址单位。

P.S.

属于计算机系统结构的属性有：数据表示、寻址方式、寄存器组织、指令系统、

存储组织、中断机构、I/O 结构、保护机构等。

属于组成的属性有：数据通路宽度、专用部件设置、功能部件并行度、控制机构的组成方式，可靠性技术等。它着眼于机器内各事件的排序方式，控制机构的功能及部件间的关系。

属于实现的属性有：部件的物理结构、器件、模块的划分与连接、微组装技术、信号传输技术等，它着眼于器件技术和微组装技术。

5、从机器（汇编）语言程序员看，以下哪些是透明的？

指令地址寄存器；指令缓冲器；时标发生器；条件寄存器；乘法器；主存地址寄存器；磁盘外设；先行进位链；移位器；通用寄存器；中断字寄存器。

答：透明的有：指令缓冲器、时标发生器、乘法器、主存地址寄存器、先进先出链、移位器

6、下列哪些对系统程序员是透明的？哪些对应用程序员是透明的？

系列机各档不同的数据通路宽度；虚拟存储器；Cache 存储器；程序状态字；“启动 I/O”指令；“执行”指令；指令缓冲寄存器。

答：对系统程序员透明的有：虚拟存储器；Cache 存储器；程序状态字；

对应用程序员透明的有：系列机各档不同的数据通路宽度；“启动 I/O”指令；“执行”指令；指令缓冲寄存器。

该题答案纯属本人个人见解。对系统程序员透明是否意味着有计算机系统结构透明，而对应用程序员透明则对应着对计算机组成透明呢？——lanjing

以下摘自南京大学出版社出版，李学干主编的辅导书

分析系统程序员是编写诸如操作系统、编译程序等各种系统软件的人员。应用程序员是指利用计算机及所配的系统软件支持来编写解决具体应用问题的程序员。他们都可以使用汇编语言或机器语言来编写程序，当然也可以用高级语言来编写程序。所以，对系统程序员或应用程序员不透明的，应包括计算机系统结构所包含的方面。而属全硬件实现的计算机组成所包含的方面，如系列机各档不同的数据通路宽度、Cache 存储器、指令缓冲寄存器等，无论是对系统程序员，还是对应用程序员都应当是透明的。对目前高性能计算机系统来讲，大多数都是多用户环境，应用程序(也称算态、目态或用户态程序)中是不允许使用管态(也称系统态、监督态)中所用的特权指令。

例如，大型多用户系统中，程序状态字是用于反映计算机系统在当前程序的各种关键状态(它并不是 IBM PC 计算机那种狭义的所谓程序状态字)，它是操作系统用于管理计算机系统资源及其使用状况的，用户不能直接对程序状态字内容进行读、写和访问的，只能由系统来管理。“启动 I/O”指令是大型机中的一种管态指令，属于特权指令，只能在操作系统程序中使用(见教材中第 3 章的 3.4.1 节所介绍)。用户程序是不能用它来直接启动 I/O 通道和设备的。虚拟存储器(参看教材第 4 章 4.1.3 节)是一个主存-辅存两级存储层次。它对应用程序是完全透明的，使应用程序不必作任何修改就可以在系统上运行。但是，在操作系统中必须配置有相应的管理软件，能对其虚实外部地址的映象和变换、程序的换道、程序由辅存调入主存、主存页面的替换、存储保护等进行管理，所以对系统程序员来说是不透明的。“执行”指令(参看教材中第 5 章 5.1.2 节)是 IBM370 等系列机上用于解决程序在执行过程中不准修改指令，又允许将指令放在操作数区中做修改，以满足指令在执行过程中允许修改的要求。这种指令无论是用户程序，还是系统程序，都希望可以使用的，所以，“执行”指令应设计成对应用程序和系统程序都是不透明的。

解答系列机各档不同数据通路宽度、Cache 存储器、指令缓冲寄存器属计算机组成，对系统程序员和应用程序员都是透明的。虚拟存储器、程序状态字、“启动 I/O”指令，对系统程序员是不透明的，而对应用程序员却是透明的。“执行”指令则对系统程序员和应用程序员都是不透明的。

7、想在系列机中发展一种新型号机器，你认为下列哪些设想是可以考虑的，哪些则不行的？为什么？

- (1) 新增加字符数据类型和若干条字符处理指令，以支持事务处理程序的编译。
- (2) 为增强中断处理功能，将中断分级由原来的 4 级增加到 5 级，并重新调整中断响应的优先次序。
- (3) 在 CPU 和主存之间增设 Cache 存储器，以克服因主存访问速率过低而造成的系统性能瓶颈。
- (4) 为解决计算误差较大，将机器中浮点数的下溢处理方法由原来的恒置“1”法，改为用 ROM 存取下溢处理结果的查表舍入法。

(5) 为增加寻址灵活性和减少平均指令字长, 将原等长操作码指令改为有 3 类不同码长的扩展操作码; 将源操作数寻址方式由操作码指明改成如 VAX-11 那种设寻址方式位字段指明。

(6) 将 CPU 与主存间的数据通路宽度由 16 位扩展成 32 位, 以加快主机内部信息的传送。

(7) 为减少公用总路线的使用冲突, 将单总线改为双总线。

(8) 把原 0 号通用寄存器改作堆栈指示器。

答: 可以考虑的有: 13467。不可以考虑的有: 258。

原则很简单, 看改进后能否保持软件的可移植性。

P.S.

为了能使软件长期稳定, 就要在相当长的时期里保证系统结构基本不变, 因此在确定系列结构时要非常慎重。其中最主要是确定好系列机的指令系统、数据表示及概念性结构。既要考虑满足应用的各种需要和发展, 又要考虑能方便地采用从低速到高速的各种组成的实现技术, 即使用复杂、昂贵的组成实现时, 也还能充分发挥该实现方法所带来的好处。

8、并行处理计算机除分布处理、MPP 和机群系统外, 有哪 4 种基本结构? 列举它们各自要解决的主要问题。

答: 除了分布处理, MPP 和机群系统外, 并行处理计算机按其基本结构特征可分为流水线计算机, 阵列处理机, 多处理机和数据流计算机四种不同的结构。

流水线计算机主要通过时间重叠, 让多个部件在时间上交叠重叠地并行招待运算和处理, 以实现时间上的并行。它主要应解决: 拥塞控制, 冲突防止, 流水线调度等问题。

阵列处理机主要通过资源重复实现空间上的并行。它主要应解决: 处理单元灵活、规律的互连模式和互连网络设计, 数据在存储器中的分布算法等问题。

多处理机主要通过资源共享, 让一组计算机在统一的操作系统全盘控制下, 实现软件和硬件各级上的相互作用, 达到时间和空间上的异步并行。它主要应解决: 处理机间互连等硬件结构, 进程间的同步步和通讯, 多处理机调度等问题。

数据流计算机设有共享变量的概念, 指令执行顺序只受指令中数据的相关性

制约。数据是以表示某一操作数或参数已准备就绪的数据令牌直接在指令之间传递。它主要应解决：研究合适的硬件组织和结构，高效执行的数据流语言等问题。

9、计算机系统的 3T 性能目标是什么？

答：计算机系统的 3T 性能目标是 1TFLOPS 计算能力，1TBYTE 主存容量和 1TBYES 的 I/O 带宽

第2章 数据表示与指令系统

1、数据结构和机器的数据表示之间是什么关系？确定和引入数据表示的基本原则是什么？

答：数据表示是能由硬件直接识别和引用的数据类型。数据结构反映各种数据元素或信息单元之间的结构关系。数据结构要通过软件映象变换成机器所具有的各种数据表示实现，所以数据表示是数据结构的组成元素。不同的数据表示可为数据结构的实现提供不同的支持，表现在实现效率和方便性不同。数据表示和数据结构是软件、硬件的界面。

除基本数据表示不可少外，高级数据表示的引入遵循以下原则：（1）看系统的效率有否提高，是否养活了实现时间和存储空间。（2）看引入这种数据表示后，其通用性和利用率是否高。

2、标志符数据表示与描述符数据表示有何区别？描述符数据表示与向量数据表示对向量数据结构所提供的支持有什么不同？

答：标志符数据表示指将数据类型与数据本身直接联系在一起，让机器中每个数所都带类型标志。其优点是：（1）简化了指令系统和程序设计；（2）简化了编译程序；（3）便于实现一致性校验；（4）能由硬件自动变换数据类型；（5）支持数据库系统的实现与数据类型无关；（6）为软件调试和应用软件开发提供支持。缺点是：（1）会增加程序所占的主存空间；（2）在微观上对机器的性能（运算速度）不利。

数据描述符指数据的描述与数据分开存放，描述所访问的数据是整块还是单个的，及访问该数据块或数据元素的地址住处它具备标志符数据表示的优点，并减少了标志符数据表示所占的空间，为向量和数组结构的实现提供支持。

数据描述符方法优于标志符数据表示，数据的描述与数据分开，描述所访问的数据是整块还是单个的，及访问该数据块或数据元素的地址信息，减少了标志符数据表示所占的篇幅。用描述符方法实现阵列数据的索引比用变址方法实现要方便，且便于检查出程序中的阵列越界错误。但它不能解决向量和数组的高速运算问题。而在有向量、数组数据表示的向量处理机上，硬件上设置有丰富的向量或阵列运算指令，配有流水或阵列方式处理的高速运算器，不仅能快速形成向量、数组的元素地址，更重要的是便于实现把向量各元素成块预取到中央处理机，用一条向量、数组指令流水或同时对整个向量、数组高速处理，如让硬件越界判断与元素运算并行。这些比起用与向量、阵列无关的机器语言和数据表示串行实现要高效的多。

3、堆栈型机器与通用寄存器型机器的主要区别是什么？堆栈型机器系统结构为程序调用的哪些操作提供了支持？

答：有堆栈数据表示的机器称为堆栈机器。它与一般通用寄存器型机器不同。通用寄存器型机器对堆栈数据结构实现的支持是较差的。表现在：(1)堆栈操作的指令少，功能单一；(2)堆栈在存储器内，访问堆栈速度低；(3)堆栈通常只用于保存于程序调用时的返回地址，少量用堆栈实现程序间的参数传递。而堆栈机器为堆栈数据结构的实现提供有力的支持。表现在：(1)有高速寄存器组成的硬件堆栈，并与主存中堆栈区在逻辑上组成整体，使堆栈的访问速度是寄存器的，容量是主存的；(2)丰富的堆栈指令可对堆栈中的数据进行各种运算和处理；(3)有力地支持高级语言的编译；(4)有力地支持子程序的嵌套和递归调用。

堆栈型机器系统结构有力地支持子程序的嵌套和递归调用。可将以下信息全部压栈，包括：保存子程序的返回地址，保存条件码，保存关键寄存器内容，保存必要的全局型、局部型参数，为子程序开辟存放局部变量和中间结果的工作区。

4、设某机阶值 6 位、尾数 48 位，阶符和数符不在其内，当尾数分别以 2、8、16 为基时，在非负阶、正尾数、规格化数情况下，求出其最小阶、最大阶、阶的个数、最小尾数值、最大尾数值、可表示的最小值和最大值及可表示的规格化数的总个数。

解：依题意知： $p=6$ $m=48$ $rm=2, 8, 16$

lanjing:

解：依题意知： $p=6$ $m'=48$ $r_m=2, 8, 16$

所求各值 r_m	2	8	16
最小阶(非负阶, 最小为0)	0	0	0
最大阶 $=2^p-1$	63	63	63
阶的个数 $=2^p$	64	64	64
最小尾数值 $=r_m^{-1}$	1/2	1/8	1/16
最大尾数值 $=1-r_m^{-m'}$	$1-2^{-48}$	$1-8^{-48}$	$1-16^{-48}$
可表示的最小值 $=r_m^{-1}$ 即 $r_m^{\text{最小阶}} \cdot \text{最小尾数值} = \text{最小尾数}$	1/2	1/8	1/16
可表示的最大值 $=r_m^{(2^p-1)} \cdot (1-r_m^{-m'})$ 即 $r_m^{\text{最大阶}} \cdot \text{最大尾数值}$	$2^{63} \cdot (1-2^{-48})$	$8^{63} \cdot (1-8^{-48})$	$16^{63} \cdot (1-16^{-48})$
可表示的规格化数的总个数 $=2^p \cdot r_m^{m'} \cdot (r_m-1)/r_m$ 即=阶的个数 \cdot 尾数个数	$2^6 \cdot 2^{47}$ $= 2^{53}$	$2^6 \cdot 7 \cdot 2^{50}$ $= 7 \cdot 2^{56}$	$2^6 \cdot 15 \cdot 2^{51}$ $= 15 \cdot 2^{57}$
可表示的尾数个数 $=r_m^{m'} \cdot (r_m-1)/r_m$	$2^{48} \cdot (2-1)/2$ $= 2^{47}$	$8^{48} \cdot (8-1)/8$ $= 7 \cdot 2^{50}$	$16^{48} \cdot (16-1)/16$ $= 15 \cdot 2^{51}$

lanjing:

5、(1) 浮点数系统使用的阶基 $r_p=2$, 阶值位数 $p=2$, 尾数基值 $r_m=10$, 以 r_m 为基的尾数位数 $m''=1$, 按

照使用的倍数来说, 等价于 $m=4$,

试计算在非负阶、正尾数、规格化情况下的最小尾数值、最大尾数值、最大阶值、可表示的最小

值和最大值及可表示数的个数。

(2) 对于 $r_p=2, p=2, r_m=4, m''=2$, 重复以上计算。

解：依题意知列下表：

$p=2, r_m=10, m''=1$ $p=2, r_m=4, m''=2$

最小尾数值 $10^{-1}=0.1$ $4^{-1}=0.25$

最大尾数值 $1-10^{-1}=0.9$ $1-4^{-2}=15/16$

最大阶值 $2p^{-1}=3$

可表示的最小值 0.1 0.25

可表示的最大值 $10^3 \times 0.9=900$ $4^3 \times 15/16=60$

可表示数的总个数 36 48

P.S

不明白题中“按照使用的倍数来说，等价于 $m=4$,”这句话是什么意思，有什么作用呢？

6、由 4 位数（其中最低位为下溢附加位）经 ROM 查表舍入法，下溢处理成 3 位结果，设计使下溢下

处理平均误差接近于零的 ROM 表，列出 ROM 编码表地址与内容的对应关系。

解：

地址 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100
1101 1110 1111

内容 000 001 001 010 010 011 011 100 100 101 101 110 110 111 111 111

7、变址寻址和基址寻址各适用于何种场合？设计一种只用 6 位地址码就可指向一个大地址空间中

任意 64 个地址之一的寻址机构。

解：基址寻址是对逻辑地址空间到物理地址空间变换的支持，以利于实现程序的动态再定位。

变址寻址是对数组等数据块运算的支持，以利于循环。

将大地址空间 64 个地址分块，用基址寄存器指出程序所在块号，用指令中 6 位地址码表示该块内 64

个地址之一，这样基址和变址相结合可访问大地址任意 64 个地址之一。

以下摘自南京大学出版社出版，李学干主编的辅导书

变址寻址适合在标量计算机中，通过循环程序访问变址寄存器，修改其变址值来对向量或数组等数据结构中的各元素进行访问和处理。基址寻址则主要是用于程序的逻辑地址空间到物理地址空间的变换，以支持程序使用动态再定位的技术。

设计一种只用 6 位地址码就可以指向一个大地址空间中任意 64 个地址之一的寻址机构，意味着指令中为寻址该操作数的地址码只有 6 位，只好用来表示这 64 个地址中的任一个。那么这 64 个地址之一应当是在哪个大的地址空间的，就得使用其他办法来指明。这里可列举常见的两种做法。

一种是采用隐含寻址，让标志这 64 个地址是相对哪个基点地址的区域用指令隐式规定的某个专门的寄存器中所存放的基址值来给出。例如，可约定某个变址寄存器或某个基址器。程序执行时，每当要访存时，就可以经硬件加法器将隐含寄存器的基址值与指令中给出的 6 位相对位移量相加来形成其访存单元的物理地址。

另一种是规定基点地址就是程序计数器，程序计数器(PC)存放的是当前所执行指令的下一条指令所在主存中的地址(或偏移地址)。因此，可以通过使用无条件转移指令来修改 PC 的内容，实现在一个大的地址空间中的访问，这就是所谓的 PC 自相对寻址。做法是通过将 PC 的内容与指令中所提供的 6 位相对位移量相加来形成主存单元的物理地址。

8.经统计，某机器 14 条指令的使用频度分别为：0.01,0.15,0.12,0.03,0.02,0.04,0.02,0.04,0.01,0.13,0.15,0.14,0.11,0.03。分别求出用等长码、Huffman 码、只有两种码长的扩展操作码 3 种编码方式的操作码平均码长。

解：

等长操作码的平均码长=4 位;Huffman 编码的平均码长=3.38 位;只有两种码长的扩展操作码的平均码长=3.4 位。

9. 若某机要求：三地址指令 4 条，单地址指令 255 条，零地址指令 16 条。设指令字长为 12 位。每个地址码长为 3 位。问能否以扩展操作码为其编码?如果其中单地址指令为 254 条呢?说明其理由。

答：①不能用扩展码为其编码。

∵指令字长 12 位，每个地址码占 3 位；

∴三地址指令最多是 $2^{(12-3-3-3)}=8$ 条， 现三地址指令需 4 条，

∴可有 4 条编码作为扩展码，

∴单地址指令最多为 $4 \times 2^3 \times 2^3 = 2^8 = 256$ 条，

现要求单地址指令 255 条， ∴可有一条编码作扩展码

∴零地址指令最多为 $1 \times 2^3 = 8$ 条

不满足题目要求

∴不可能以扩展码为其编码。

②若单地址指令 254 条，可以用扩展码为其编码。

∴依据①中推导，单地址指令中可用 2 条编码作为扩展码

∴零地址指令为 $2 \times 2^3 = 16$ 条，满足题目要求

10、某机指令字长 16 位。设有单地址指令和双地址指令两类。若每个地址字段为 6 位。

且双地址指

令有 X 条。问单地址指令最多可以有多少条？

答：

双地址指令最多是 $2^{(16-6-6)} = 2^4 = 16$ 条， 现双地址指令有 X 条，

∴可有 $(16-X)$ 条编码作为扩展码，

∴单地址指令最多为 $(16-X) \times 2^6$

11. 何谓指令格式的优化？简要列举包括操作码和地址码两部分的指令格式优化可采用的各种途径

和思路。

答：指令格式的优化指如何用最短位数表示指令的操作信息和地址信息，使程序中指令的平均字

长最短。

为此用到 Huffman 压缩概念。其基本思想是，当各种事件发生概率不均等时，采用优化技术对

发生概率最高的事件用最短的位数(时间)来表示(处理)，而对出现概率较低的事件，允许采用较

长位数(时间)来表示(处理)，会导致表示(处理)的平均位数(时间)的缩短。

①用此思想可对操作码进行优化。首先通过大量已有典型程序进行统计，可得到每种指令在

程序中出现的概率(使用频度)。然后构造它们的 Huffman 树。方法如下：

a)被统计指令按使用频由小到大排列；

b)每次选择其中最小的二个频度合成一个频度是它们二者之和的新结点，并将该结点按频度

大小插到余下的未参与结合的频度值中；

c)如此继续，直至全部频度结合完毕形成根结点。

最后从根结点开始对每个结点的两个分支分别用 0、1 表示，则到达各频度指令的代码序列就构成该频度指令的 Huffman 码。

Huffman 码是最优化的编码，但这种编码码长种类太多，不便于译码，不能实用，为此可采用

下面的扩展操作码编码。扩展操作码编码是介于定长二进制编码和全 Huffman 编码之间的一种编

码，仍利用 Huffman 压缩思想，使操作码平均长度缩短。其操作码长度不定，但只有有限几种码

长，是一种可实用的优化编码方法。扩展方法应根据指令使用频度 p_i 的分布而定。如 p_i 值在头 15

种指令中都比较长，但在 30 种指令以后急剧减少，则宜选 15/15/15 法；若 p_i 值在头 8 种指令中较

大，之后的 64 种指令 p_i 值也不低时，则宜选 8/64/512 法。衡量标准是哪种编码使平均长度最短。

②对地址码的优化：

操作码的优化表示可以使指令总位数减少，但不降低访存指令的速度，必须维持指令字按

整数边界存储，所以首先应考虑地址码也用可变长，让长操作码与短地址码配合，即使是定长指

令字，也可利用操作码优化腾出的空白。减少存储空间的浪费。如果最常用的操作码最短，其地

址码个数越多指令功能越强。如为实现 $A+B \rightarrow C$ ，若采用单地址指令需经取 A，加 B，送 C 三条指令

完成，而用了三地址指令只需一条，减少程序占用空间。

其次考虑多种寻址方式在满足很大寻址空间前提下，缩短地址码位数。如在 IBM370 中用 16 位

地址码可通过基址寻址形成 24 位访存物理地址。以 20 位地址码通过基址+变址形成 24 位访存物理地

址。还可采用相对寻址，分段存储管理方式。

另外可考虑在同种地址制下的多种地址形式。如让空白处放直接操作数或常数。

12.某模型机 9 条指令使用频率为：

ADD(加) 30% SUB(减) 24% JOM(按负转移) 6%

STO(存) 7% JMP(转移) 7% SHR(右移) 2%

CIL(循环) 3% CLA(清加) 20% STP(停机) 1%

要求有两种指令字长，都按双操作数指令格式编排，采用扩展操作码，并限制只能有两种操作码码长。设该机有若干通用寄存器，主存为 16 位宽，按字节编址，采用按整数边界存储。

任何指令都在一个主存周期中取得，短指令为寄存器-寄存器型，长指令为寄存器-主存型，主存地址应能变址寻址。

(1)仅根据使用频率，不考虑其它要求，设计出全 Huffman 操作码，计算其平均码长；

(2)考虑题目全部要求，设计优化实用的操作形式，并计算其操作码的平均码长；

(3)该机允许使用多少可编址的通用寄存器？

(4)画出该机两种指令字格式，标出各字段之位长；

(5)指出访存操作数地址寻址的最大相对位移量为多少个字节？

解：

第(1)和(2)中 Huffman 和扩展操作码的编码及平均码长如下表：

指令 i 使用频度 P_i Huffman 编码 扩展操作码编码

I1 30% 10 00

I2 24% 00 01

I3 20% 01 10

I4 7% 1100 11000

I5 7% 1101 11001

I6 6% 1110 11010

I7 3% 11110 11011

I8 2% 111110 11100

I9 1% 111111 11101

西个马 p_i 2.61 2.78

(3)8 个。

(4)两种指令格式如下图所示：

2 位 3 位 3 位

OP R1 R2

操作码 寄存器 1 寄存器 2

5 位 3 位 3 位 5 位

OP R1 X d

操作码 寄存器 1 变址寄存器 相对位移

主存逻辑地址

(5)访存操作数地址寻址的最大相对位移量为 32 个字节。

13. 设计 RISC 机器的一般原则及可采用的基本技术有那些？

答：一般原则：(1)确定指令系统时，只选择使用频度很高的指令及少量有效支持操作系统，高级

语言及其它功能的指令；(2)减少寻址方式种类，一般不超过两种；(3)让所有指令在一个机器周

期内完成；(4)扩大通用寄存器个数，一般不少于 32 个，尽量减少访存次数；(5)大多数指令用硬

联实现，少数用微程序实现；(6)优化编译程序，简单有效地支持高级语言实现。

基本技术：(1)按 RISC 一般原则设计，即确定指令系统时，选最常用基本指令，附以少数对操

作系统等支持最有用的指令，使指令精简。编码规整，寻址方式种类减少到 1、2 种。(2)逻辑实现

用硬联和微程序相结合。即大多数简单指令用硬联方式实现，功能复杂的指令用微程序实现。

(3)

用重叠寄存器窗口。即：为了减少访存，减化寻址方式和指令格式，简单有效地支持高级语言中

的过程调用，在 RISC 机器中设有大量寄存器，并让各过程的寄存器窗口部分重叠。(4)用流水和延

迟转移实现指令，即可让本条指令执行与下条指令预取在时间上重叠。另外，将转移指令与其前

面的一条指令对换位置，让成功转移总是在紧跟的指令执行之后发生，使预取指令不作废，节省

一个机器周期。(5)优化设计编译系统。即尽力优化寄存器分配，减少访存次数。不仅要利

用常规

手段优化编译，还可调整指令执行顺序，以尽量减少机器周期等。

14. 简要比较 CISC 机器和 RISC 机器各自的结构特点，它们分别存在哪些不足和问题？

为什么说今后

的发展应是 CISC 和 RISC 的结合？

答：CISC 结构特点：机器指令系统庞大复杂。

RISC 结构特点：机器指令系统简单，规模小，复杂度低。

CISC 的问题：

- (1)指令系统庞大，一般 200 条以上；
- (2)指令操作繁杂，执行速度很低；
- (3)难以优化生成高效机器语言程序，编译也太长，太复杂；
- (4)由于指令系统庞大，指令的使用频度不高，降低系统性能价格比，增加设计人员负担。

RISC 的问题：

- (1)由于指令少，在原 CISC 上一条指令完成的功能现在需多条 RISC 指令才能完成，加重汇编语言程序设计负担，增加了机器语言程序长度，加大指令信息流量。

(2)对浮点运算和虚拟存储支持不很强。

(3)RISC 编译程序比 CISC 难写。

由于 RISC 和 CISC 各有优缺点，在设计时，应向着两者结合，取长补短方向发展。

The skyline:

单地址指令格式： 指令 地址

9 3 位

所以前面 9 位由于三地址指令用了最前面 3 位，还有中间 6 位可作为编码（也就是总共可以有 9 位作为单地址指令的指令操作码的编码）。减去 3 地址指令的 4 条，有 $4 \times 2^6 = 256$ 条，

但由于题目要求要有 255 条，所以剩下一个编码，已经用了 9 位的全部编码，最后零地址指令（全部 12 位都可作为操作码的编码）还有 $1 \times 2^3 = 8$

（这是 12 位编码中最后三位的）若只要求 254 种，则可以有 $(256 - 254) \times 2^3 = 16$ 条

第3章 总线、中断与输入输出系统

3.1. 简要举出集中式串行链接，定时查询和独立请求3种总线控制方式的优缺点。同时分析硬件产生故障时通讯的可靠性。

答：集中式串行链连接方式。其过程为：

①所有部件都经公共的“总线请求”线向总线控制器发使用总线申请。

②当“总线忙”信号未建立时，“总线请求”才被总线控制器响应，送出“总线可用”信号，它串行地通过每个部件。

③如果某部件未发过“总线请求”，则它将“总线可用”信号往下一部件转，如果某部件发过“总线请求”，则停止“总线可用”信号的传送。

④该部件建立“总线忙”，并除去“总线请求”，此时该部件获得总线使用权，准备传送数据。

⑤数据传送期间，“总线忙”维持“总线可用”的建立。

⑥传送完成后，该部件去除“总线忙”信号和“总线可用”信号。

⑦当“总线请求”再次建立时，就开始新的总线分配过程。

优点：①选择算法简单；②控制总线数少；③可扩充性好；④可靠性高。

缺点：①对“总线可用”线及其有关电路失效敏感，②不灵活；③总线中信号传送速度慢。

集中式定时查询方式，过程：

①总线上每个部件通过“总线请求”发请求。

②若“总线忙”信号未建立，则计数器开始计数，定时查询个部件，以确定是谁发的请求。

③当查询线上的计数值与发出请求的部件号一致时，该部件建立“总线忙”，计数停止，查询也停止。除去“总线请求”，该部件获得总线使用权。

④“总线忙”维持到数据传送完毕。

⑤数据传送完，去除“总线忙”。

⑥当“总线请求”线上有新的请求，就开始下一个总线分配过程。

优点：①优先次序灵活性强；②可靠性高。

缺点：①控制线数较多；②扩展性较差；③控制较为复杂；④总线分配受限于计数信号，不能很高。

集中式独立请求方式，过程：

①每个部件有一对“总线请求”和“总线准许”线。

②每个部件使用“总线请求”发申请，当“总线已分配”无信号时，总线控制器根据某种算法对同时送来的多个请求进行仲裁，以确定哪个部件使用总线，信号从“总线准许”送回该部件，去除该部件的“总线请求”，建立总线已分配”。

③获得总线使用权的部件传送数据，直至完毕。

④数据传送完毕后，除去总线已分配”和“总线准许”，开始新的总线分配。

优点：①总线分配速度快；②灵活；③能方便隔离失效部件的请求。

缺点：①控制线数多；②复杂。

硬件产生故障时，通信的可靠性分析：

串行链接对通信的可靠性，主要表现在“总线可用”线及其有关电路的失效会造成系统整体瘫痪的问题。一旦“总线可用”线出现断路或碰地，其高电平不能顺链往下传送，就会使后面的部件在要求使用总线时，其请求无法得到响应。为了提高可靠性，可对“总线可用”线及其相关电路，采取重复设置两套或多套来解决。

定时查询对通信的可靠性要求比串行链接的高。因为总线控制器通过计数，查询到某个出故障的部件时，故障部件不会给出“总线忙”信号，这样，不会影响控制器继续计数，去查询下一个部件，所以整个总线系统的工作不会瘫痪。

独立请求对通信的可靠性要求同样比串行链接的高。因为某个部件在发生故障时，不发总线请求，即使发出总线请求，总线控制器也可以通过软件或硬件的措施，将发生故障的部件送来的请求予以屏蔽掉，不让其参与总线的分配。所以，某个部件的故障不会导致整个系统的工作处于瘫痪状态。

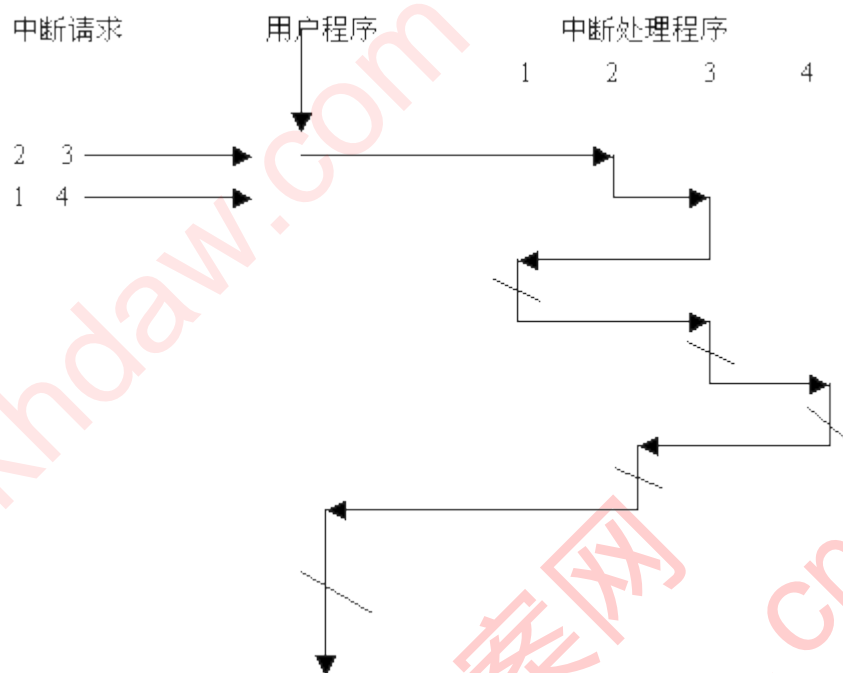
3.2. 设中断级屏蔽位“1”对应于开放，“0”对应于屏蔽，各级中断处理程序的中断级屏蔽位设置如下：（见课本）

(1)当中断响应优先次序为 1→2→3→4 时，其中断处理次序是什么？

答：(1)1—3—4—2 中断处理程序

(2)如果所有的中断处理都各需 3 个单位时间，中断响应和中断返回时间相对中断处理时间少得多。当机器正在运行用户程序时，同时发生第 2，3 级中断请求，过两个单位时间，又同时发生第 1，4 级中断请求，试画出程序运行过程示意图。

答：



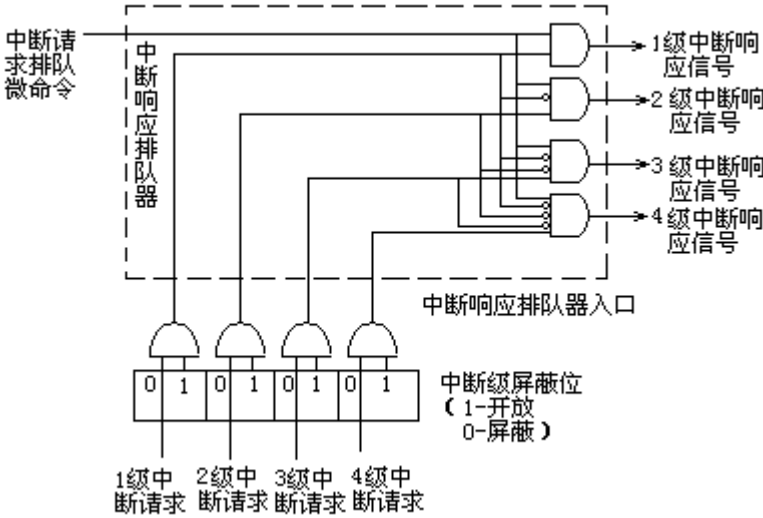
《南京大学出版社》李学干主编的辅导书 对本题的分析：

为了领会中断响应排队器对中断响应优先次序是用硬件固定的，以及通过由操作系统给各中断级服务程序现行程序状态字中的中断级屏蔽位设置不同的状态，可以改变中断处理(完)的次序这两个要点，下图给出了一个中断响应硬件部分的简单逻辑原理示意图。图中略去了某些实现上的具体细节，因为这些已不是本课程要讨论的内容。

中断级屏蔽位是程序状态字中的一个组成部分。程序状态字是将散布于系统各部分，反映程序工作时某些关键性硬件的状态，组合在一起所构成的字，有的计算机也称其为处理器状态字或程序换道区。每类程序均在主存中指定一个区域来放置其程序状态字。运行一个程序或进程时，就会将其程序状态字从主存指定单元或区域取出送到分散于系统各部分的寄存器或计数器中，建立起运行此程序或进程的环境。一个程序或进程在退出运行时，也会将反映该程序状态的这些寄存器或计数器内容组拼成程序状态字，存回该程序或进程在主存中的指定单元或区域里。因此，程序或进程的切换，只需要通过硬件启动的交换新旧程序状态字的内容即可快速完成。例如，在 IBM370 系列机上，程序状态字为 64 位，等于它的长字，交换程序状态字只需硬件启动写长字和读长字两次访存即可完成。

尽管中断请求是随机发出的，为了便于精确保存中断的断点以及在中断处理完后

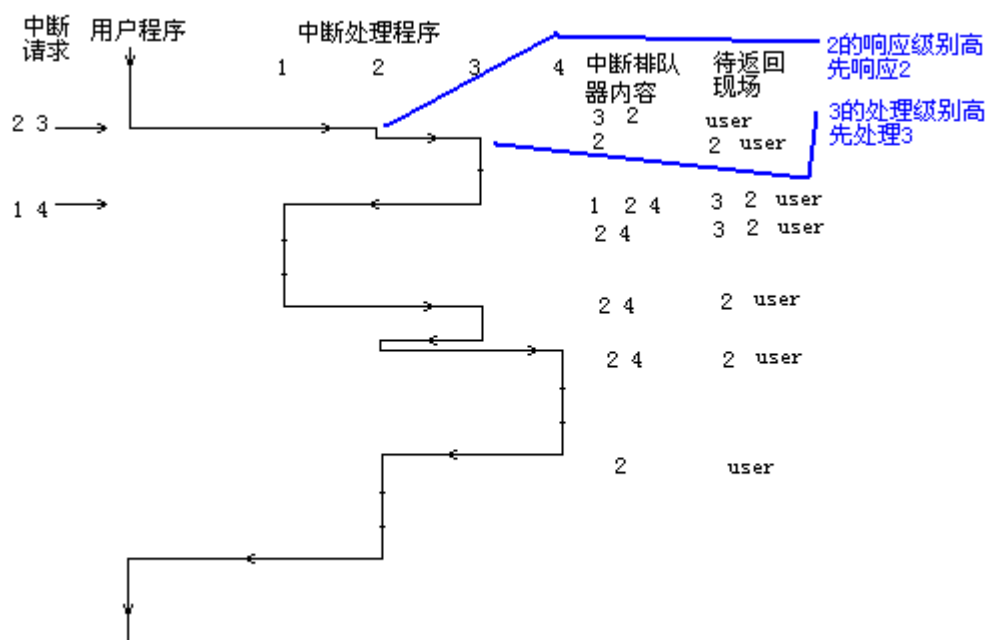
又能返回到原中断处，中断响应排队器总是在每条指令执行到最后一个机器周期的最后一个时钟周期时，对目前到达中断响应排队器入口的所有中断请求排一次队，择优进行响应。在中断响应排队器相应的输出端产生出响应信号。此信号经中断级服务程序入口地址形成硬件，生成出该级中断服务程序的程序状态字在内存区中所存放的地址。同时，经中断响应控制信号启动，进行新旧程序状态字的交换，完成程序的切换。被中断的程序的断点地址(即程序计数器的内容)，由硬件自动压入返回地址堆栈，予以保存。系统切换到新的程序或进程后，继续运行下去。如果新的程序或进程是一个中断服务程序，在运行结束，执行到中断返回指令时，就会从堆栈中弹出所保存的返回地址，再次交换程序状态字，系统又重新返回到原被中断的程序，恢复运行。



当然，低级的中断服务程序在处理过程中又遇到了比其更高级的中断请求时，应允许其被中断，以实现多级中断的嵌套。利用返回地址堆栈的后进先出工作方式，就可以完成中断嵌套时的正确返回。可以看出，只要某道程序运行时，由操作系统在现行程序的程序状态字中，根据对各中断级的中断请求是否屏蔽，设置好中断级屏蔽位的状态，就可以控制这些级别的中断请求是否进入中断响应排队器去参加排队。只有能进入中断响应排队器的中断级请求，才有机会得到响应，从而就可改变中断实际处理完的次序。

应当注意的是，用户程序是不能屏蔽任何中断的。因此，用户程序的现行程序状态字中，对各中断级的屏蔽位均应让其处于“开放”状态。

根据本题中所给出的各级中断处理程序对中断级屏蔽位设置的状况,很容易得出其中断处理(完)的次序应当是 **1342**。因为正处理 **1** 级的中断处理程序时,现行程序状态字中的中断级屏蔽位为 **0000**,在其执行期间,任何新的同级和低级的中断请求都不可能进入中断响应排队器进行排队,所以, **1** 级中断处理程序一定会先处理完。当执行 **3** 级中断服务程序时,由于现行程序状态字中的中断级屏蔽位为 **1000**,即对 **1** 级中断请求是“开放”的,而对其他各级中断请求则处于“屏蔽”状态。所以,只要此时发生 **1** 级中断请求,它就能进入中断响应排队器去排队。从而在中断请求排队的微操作发出时,就可打断 **3** 级中断服务程序的执行,交换程序状态字,转去执行 **1** 级中断处理程序,使之被优先处理完。而在执行 **3** 级中断服务程序时,由于现行程序状态字对 **2、3、4** 级的中断请求处于被“屏蔽”的状态,所以,它们都不能打断正在执行的 **3** 级中断处理程序。其他的情况可以依此类推得到。

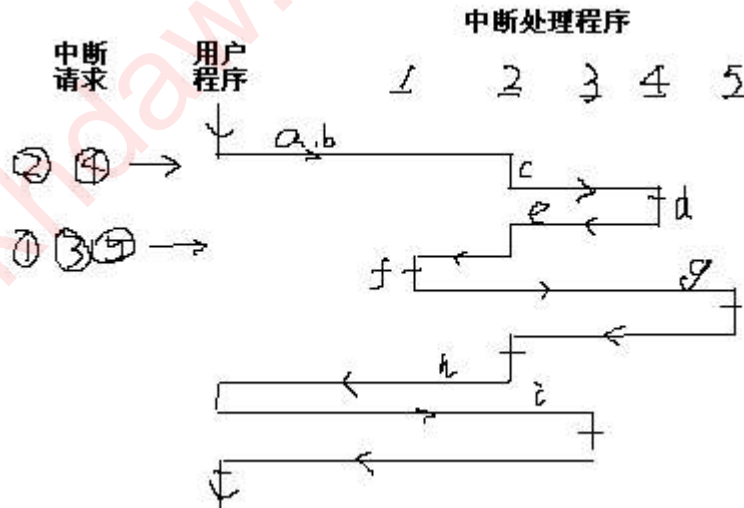


3.3.若机器共有 **5** 级中断,中断响应优先次序为 **1→2→3→4→5**,现要求其实际的中断处理次序 **1→4→5→2→3**。

(1)设计各级中断处理程序的中断级屏蔽位(令“**1**”对应于开放,“**0**”对应于屏蔽);
略

(2)若在运行用户程序时，同时出现第 4，2 级中断请求，而在处理第 2 级中断未完成时，又同时出现第 1，3，5 级中断请求，请画出此程序运行过程示意图。

答：(选自老版主的答案)



1) 五个级别的中断屏蔽位分别为 (1 开放; 0 屏蔽) :

1: 00000 2: 10011 3: 11011 4: 10000 5: 10010

2) 中断过程示意图: 如图

a. 2、4 中断同时出现，进行排队器；

b. 按中断响应优先次序，2 响应；

c. 此时屏蔽字为 10011，所以；

d. 响应 4，中断 4 运行结束，回 2；

e. 1、3、5 进入排队器，此时屏蔽字为 10011，且 1 优先级最高，所以；

f. 响应 1，1 运行结束，回 2，根据屏蔽字，所以；

g. 5 响应，5 运行结束，回 2；

h. 根据屏蔽字，不响应 3，所以 2 运行结束；回用户程序；

i. 3 还在排队器，响应 3，运行直到结束，回用户程序

3.4. 简述字节多路，数组多路和选择通道的数据传送方式。

答：字节多路通道适用于连接大量的字符类低速设备。它以字节交叉方式轮流为多台设备服务，它可有多个子通道，它们分时进入通道。

数组多路通道适合于连接多台高速设备，每传送一个定长块就选择一次设备，多

台设备以成组交叉方式工作。它可有多个子通道。它们分时进入通道。

选择通道方式适合于优先级高的高速设备，让它独占通道，数据传送以不定长方式进行，在数据传送期只选择一次设备。

3.5 如果通道在数据传送期中，选择设备需 $9.8\mu\text{s}$ ，传送一个字节数据需 $0.2\mu\text{s}$ 。某低速设备每隔 $500\mu\text{s}$ 发出一个字节数据传送请求，问至多可接几台这种低速设备？对于如下 A~F 种高速设备，一次通讯传送的字节数不少于 1024 个字节，问哪些设备可以挂在此通道上？哪些则不能？其中 A—F 设备每发出一个字节数据传送请求的时间间隔分别为(单位为 μs)：

设备 A B C D E F

发申请间隔 0.2 0.25 0.5 0.19 0.4 0.21

答：

(1) ∵ 选择设备需 $9.8\mu\text{s}$ ，传送一个字节需 $0.2\mu\text{s}$

∴ 该通道完成一个字节的数据需 $9.8+0.2=1\mu\text{s}$

∴ 某低速设备每隔 $500\mu\text{s}$ 发出一字节数据请求，为使数据不丢失，该通道可连设备数至多为 $500\mu\text{s}/1\mu\text{s}=500$ 台。

(2) 对于高速设备，由于一次传送字节数不少于 1024byte

∴ 该通道一次传送数据的时间为 $9.8\mu\text{s}+1024\times 0.2\mu\text{s}=214.6\mu\text{s}$

由表中可得出每台设备发送 1024 字节的时间间隔分别为

A B C D E F

单位 μs 204.8 256 512 194.56 409.6 215.04

∴ 为使数据不丢失，B、C、E、F 可挂在该通道上。A、D 不能。

3.6 某字节多路通道连接 6 台外设，某数据传送速率分别如表中所列。

设备 1 2 3 4 5 6

传送速率 (KB/s) 50 15 100 25 40 20

(1) 计算所有设备都工作时的通道实际最大流量： 答：实际最大流量 = $50+15+100+25+40+20=250\text{KB/S}$ 。

(2) 如果设计的通道工作周期使通道极限流量恰好与通道最大流量相等，以满足流量设计的基本要求，同时让速率越高的设备被响应的优先级越高。当 6 台设备同时发出请求开始，画出此通道在数据传送期内响应和处理各外设请求的时间

示意图。由此你发现了什么问题？

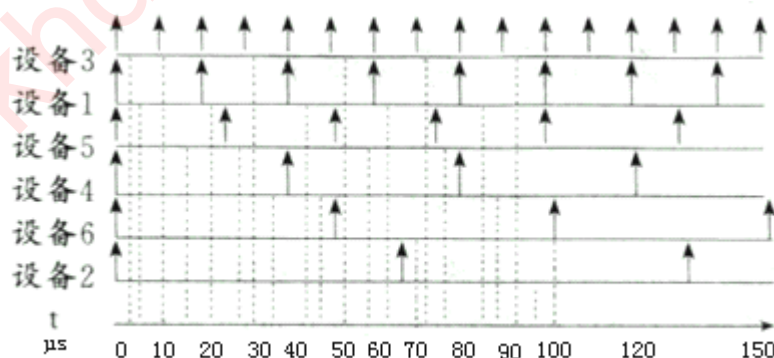
答：由表可解各设备连续发送两个字节的时间间隔分别为：

1 2 3 4 5 6

20 μ s 67 μ s 10 μ s 40 μ s 25 μ s 50 μ s

KB=1024B, $s=10^6\mu$ s, 设备 1 的时间间隔为 $10^6/(50*1024)\approx 20\mu$ s, 其他如同 1。

为简化计算, 可视 1024 为 1000



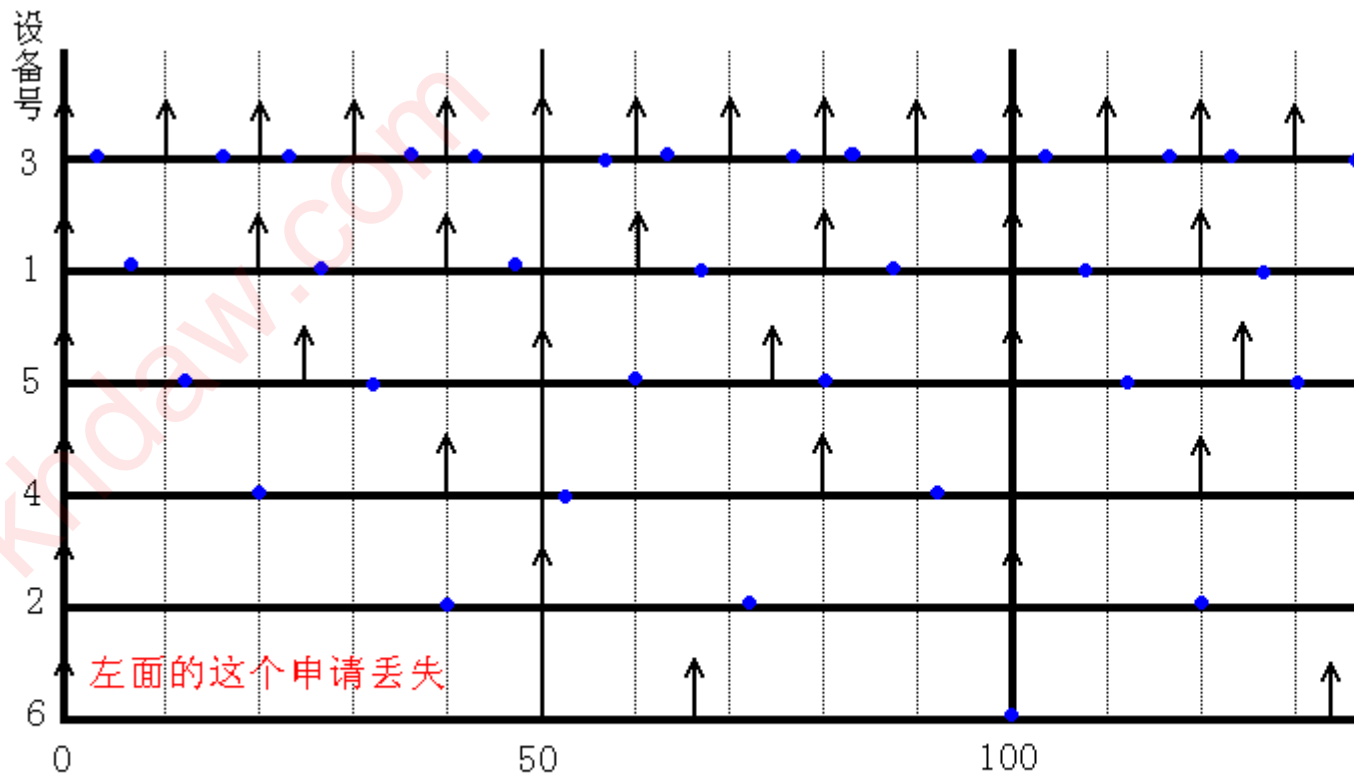
由此发现由于高速设备的响应优先级高, 使低速设备 6 和设备 2 造成数据丢失。

(3)在(2)的基础上, 在哪台设备内设置多少个字节的缓冲器就可以避免设备信息丢失?那么, 这是否说书中关于流量设计的基本要求是没有必要的了呢?为什么?

答：在设备 6 和 2 中各设两个字节的缓冲区即可。

这并不说明流量设计的基本条件是不必要的, 因为若基本条件不满足, 无论设备优先级如何确定总有设备的信息会丢失。

阿利：这是我画的通道处理设备请求的示意图。（请点击看全图）



↑ 表示设备提出申请的时刻

• 表示通道处理完设备申请的时刻

由图中可看出，6号设备有丢失信息现象，解决的办法是在6号设备内设置2个字节的缓冲器。

3.7 通道型 I/O 系统由一个字节多路通道 A(其中包括两个子通道 A1 和 A2)，两个数组多路通道 B1 和 B2 及一个选择通道 C 构成，各通道所接设备和设备的数据传送速率如表所示。（见课本）

(1) 分别求出各通道应具有多大设计流量才不会丢失信息；

答：子通道 A1 的最大实际流量 = $50 + 35 + 20 + 20 + 50 + 35 + 20 + 20 = 250 \text{ KB/S} = 0.25 \text{ MB/S}$

25MB/S

∴ 子通道 A1 至少应有 0.25MB/S 的流量才不丢失信息。

同理子通道 A2 的流量必须 $\geq 0.25 \text{ MB/S}$ 子通道 B1 的实际最大流量 = 0.5 MB/S

∴ B1 流量至少为 0.5MB/S。

同理子通道 B2 流量至少设计成 0.5MB/S。

选择通道 C 的流量至少设计成 0.5MB/S。

(2) 设 I/O 系统流量占主存流量的 $1/2$ 时才算流量平衡, 则主存流量应达到多少?

答: 此 I/O 系统的流量应为各子通道流量之和。即为 $0.25+0.25+0.5+0.5+0.5=2\text{MB/S}$

依题意 I/O 系统流量占主存流量的 $1/2$ 才算流量平衡。

因此主存流量应达到 4MB/S 。

第四章课后题

1、设二级虚拟存储器的 $TA1=10^{-7}s$ 、 $TA2=10^{-2}s$,为使存储层次的访问效率 e 达到最大值的 80%以上,命中率 H 至少要求达到多少?实际上这样高的命中率是很难达到的,那么从存储层次上如何改进?

解: $\because e=1/[H+(1-H)r]$ 且 $r=TA2/TA1 \therefore H$ 至少达到 99.9%

这样的命中率很难达到,可在二级存储器间加一层电子磁盘,降低 r ,从而降低对 H 的要求。

2、程序存放在模 32 单字交叉存储器中,设访存申请队的转移概率 λ 为 25%,求每个存储周期能访问到的平均字数。当模数为 16 呢?由此你可得到什么结论?

解: $B=[1-(1-\lambda)^m]/\lambda$

由 $\lambda=0.25, m=32$ 求得: $B=4-4*(3/4)^{32}=4$

同理, $m=16$ 时, $B=4-4*(3/4)^{16}=3.96$

由此可看出,当转移概率 λ 为 25%比较大时,采用模 32 与模 16 的每个存储周期能访问的平均字数非常相近。就是说,此时,提高模数 m 对提高主存实际频宽已不显著。实际上,模数 m 的进一步增大,会因工程实现上的问题,导致实际性能反而可能比模 16 的还要低,且价格更高。所以模数 m 不宜太大。对于 λ 为 25%的情况,可以计算出 $m=8$ 时,其 B 已经接近于 3.6 了。

3、设主存每个分体的存取周期为 $2\mu s$,宽度为 4 个字节。采用模 m 多分体交叉存取,但实际频宽只能达到最大频宽的 0.6 倍。现要求主存实际频宽为 4MB/S,问主存模数 m 应取多少方能使两者速度基本适配?其中 m 取 2 的幂。

解: 由题意已知存取周期 $T_m=2*10^{-6}s$,宽度 $W=4B$, $B_{实}=0.6B_m=4*2^{20}B/S$,

$B_m=W*m/T_m=6.99*10^6B/S$

$m=B_m*T_m/W=6.99*10^6*2*10^{-6}/4=3.495$

所以 m 取 4 能满足要求

P.S.

①微秒(百万分之一秒) $1\mu s=10^{-6}s$

②计量单位中的 M(兆)是 10 的 6 次方,见到 M 自然想起要在该数值的后边续上六个 0,即扩大一百万倍。在二进制中,MB 也表示到了百万级的数量级,但 1

MB 不正好等于 1000000 字节, 而是 1048576 字节, 即 $1\text{MB} = 2^E+20 \text{ Bytes}$
= 1048576Bytes。

4、某虚拟存储器共 8 个页面, 每页 1024 个字, 实际主存为 4096 个字, 采用页表法进行地址映象。映象表的内容如下表 1 所示。

实页号 装入位

3 1

1 1

2 0

3 0

2 1

1 0

0 1

0 0

表 1

虚页号 实页号 装入位

0 3 1

1 1 1

2 2 0

3 3 0

4 2 1

5 1 0

6 0 1

7 0 0

表 2

(1)列出会发生页面失效的全部虚页号;

解: 根据页表法列出表 2, 当装入位为 0 时, 即为页面失效, 再找出相对应的虚页号即可。

会发生页面失效的全部虚页号为: 2,3,5,7

(2)按以下虚地址计算主存实地址: 0, 3728, 1023, 1024, 2055, 7800, 409

6, 6800。

解：虚页号= $\lfloor \text{虚地址} / \text{页面大小} \rfloor$

实地址=(实页号*页面大小)+(虚地址-虚页号*页面大小)

虚地址 0 3728 1023 1024 2055 7800 4096 6800

虚页号 0 3 0 1 2 7 4 6

实页号 3 3 3 1 2 0 2 0

装入位 1 0 1 1 0 0 1 1

实地址 3072 3728 4095 1024 2055 632 2048 656

5、一个段页式虚拟存储器。虚地址有 2 位段号、2 位页号、11 位页内位移（按字编址），主存容量为 32K 字。每段可有访问方式保护，其页表和保护位如下表所示。

(1) 此地址空间中共有多少个虚页？

解： $2N_v$ 页，而 $N_v = \text{用户虚页号} = \text{段号 } S + \text{页号 } P$

此地址空间中共有 $2^N v = 2^{(2+2)} = 16$ 个虚页

(2) 当程序中遇到下列各情况时，写出由虚地址计算出的实地址。说明哪个会发生段失效、页失效

方式	段	页	页内位移
取数	0	1	1
取数	1	1	10
取数	3	3	2047
存数	0	1	4
存数	2	1	2
存数	1	0	14
转移至此	1	3	100
取数	0	2	50
取数	2	0	5
转移至此	3	0	60

表1

方式	段	页	页内位移	实页号	实地址/失效状态
取数	0	1	1	3	$3 \times 2 \times 1024 + 1 = 6145$
取数	1	1	10	0	$2 \times 2K + 10 = 10$
取数	3	3	2047	指令/数据在辅存	页失效
存数	0	1	4	3	保护失效
存数	2	1	2	页表不在主存	段失效
存数	1	0	14	指令/数据在辅存	页失效
转移至此	1	3	100	8	$8 \times 2K + 100 = 16484$
取数	0	2	50	指令/数据在辅存	页失效
取数	2	0	5	页表不在主存	段失效
转移至此	3	0	60	14	保护失效

表2

解：由题意“11位页内位移（按字编址）”知：页大小为2K字

P. S. 一个字节为8位，一个字为两个字节。若按节编址，11位就要占2个字。又因为主存容量为32K字

A. 第一种情况：段0页1根据页表查得对应实页号为实页3，所以地址为 $3 \times 2K + 1 = 3 \times 2 \times 1024 + 1 =$

6、设某程序包含 5 个虚页，其页地址为 4, 5, 3, 2, 5, 1, 3, 2, 2, 5,

1, 3。当使用 LRU 算法替

换时，为获得最高命中率，至少应分配给该程序几个实页？其可能的最高命中率

为多少？

khdaw.com

课后答案网
www.hackshp.cn

khdaw.com

解：(I) 当实页数为1时

时间t	1	2	3	4	5	6	7	8	9	10
页地址流	4	5	3	2	5	1	3	2	2	5
命中1次	4*	5*	3*	2*	5*	1*	3*	2*	2*	5*
	调进	替换	替换	替换	替换	替换	替换	替换	命中	替换

(II) 当实页数为2时

时间t	1	2	3	4	5	6	7	8	9	10
页地址流	4	5	3	2	5	1	3	2	2	5
命中1次	①4	②4*	①3	②3*	①5	②5*	①3	②3*	②3*	①5
	/	①5	②5*	①2	②2*	①1	②1*	①2	①2	②2*
	调进	调进	替换	替换	替换	替换	替换	替换	命中	替换

(III) 当实页数为3时

时间t	1	2	3	4	5	6	7	8	9	10
页地址流	4	5	3	2	5	1	3	2	2	5
命中2次	①4	②4	③4*	①2	②2	③2*	①3	②3	②3	③3*
	/	①5	②5	③5*	①5	②5	③5*	①2	①2	②2
	/	/	①3	②3	③3*	①1	②1	③1*	③1*	①5
	调进	调进	调进	替换	命中	替换	替换	替换	命中	替换

(IV) 当实页数为4时

时间t	1	2	3	4	5	6	7	8	9	10
页地址流	4	5	3	2	5	1	3	2	2	5
命中7次	①4	②4	③4	④4*	④4*	①1	②1	1	1	1
	/	①5	②5	③5	①5	②5	③5	5	5	5
	/	/	①3	②3	③3	④3*	①3	3	3	3
	/	/	/	①2	②2	③2	④2*	2	2	2

4-6. 采用堆栈模拟, 分析实页数为 3, 4, 5 时的命中率:

堆栈法参考 P.116 页中的说明.

当 $n=4$ 时

地址流	4	5	3	2	5	1	3	2	2	5	1	3
堆栈中	4	5	3	2	5	1	3	2	2	5	1	3
		4	5	3	2	5	1	3	3	2	5	1
			4	5	3	2	5	1	1	3	2	5
				4	4	3	2	5	5	1	3	2
					4	4	4	4	4	4	4	4

实页数 $n=$

$n=3$

H

H

$n=4$

H

H

H

H

H

H

$n=5$

H

H

H

H

H

H

当 $n=4$ 时, H 已最大. 故 $H = 6/12 = 1/2$.

4-4 题: 比较重要, 注意弄懂.

分析: 虚页 8 页. 实页 4 页: $4096/1024 = 4$ 页

页表法即/全相联映像. 右边表格中已有对应映像表: $0 \rightarrow 3$ (前直后实). 表示虚页 0

装入主存. 举例: 第 3 个数: 1023. 它是虚页第 0 页, 页内位移 1023. 对应实页 3. 即 3

再加上 1023 的偏移 即实页地址: 4095.

4-5 分析: 虚页: $2(\text{段}) + 2(\text{页}) = 4$ 位, 即 $2^4 = 16$ 个虚页.

实页: 页大小: 2^{11} (与虚页大小相同). 容量 $32K = 2^{15} = 2^{11} \times 2^4$ 字. 实页数 $2^{15}/2^{11} = 2^4 = 16$ 页.

举例: <A> 第一行: 取数, 段 0 页 1, 查上表有 实页 3 (只读, 取数即/即是读).

实页 3 的起始地址: 页大小 \times (实页数): 2^{11} 就是 2048.

$$2048 \times 3 = 6144$$

再加上页内位移 1, 有 $6144 + 1 = 6145$. : 结果: 实页

 第 7 行: 转移到此段 1 页 3. 查得 实页 8 (可读, 执行): 转移到此

4-8: 1K 个任务, 即用户数为 1024.

$$2048 \times 8 + \text{位移 } 100 = 16484$$

10 位用户号

虚地址: 任务号 (10 位) + 虚页号 12 位 (4096 页) + 页位移 12 位 (512 页 $2^9 \times 2^3$)

实地址: 实页号 8 位 ($2^{20}/2^{12}$) + 页位移 12 位 (512 页)

解答: (1) 不画了. 参考书 4-21 图.

(2) 相联比较位数: 比较的是任务号, 即 10 位 (1K 个用户任务)

(3) 相联每个寄存器位数: 10 位 + 2 号 (ID 号, 4 个任务运行, 2^2 个 ID)

(4) 散列变换硬件 输入位数: ID 号 (2 位) + 虚页号 12 位 = 14 位

输出位数: 快表地址. 5 位 ($2^5 = 32$ 行)

14 -

(5) 每个相联比较器的位数: ID (2 位) + 虚页号 12 位 = 14 位

7.采用页式管理的虚拟存储器，分时运行两道程序。其中，程序 X 为

```
DO 50 I=1,3
```

```
  B(I)=A(I)-C(I)
```

```
  IF(B(I)·LE·0)GOTO 40
```

```
  D(I)=2*C(I)-A(I)
```

```
  IF(D(I)·EQ·0)GOTO 50
```

```
40  E(I)=0
```

```
50  CONTINUE
```

```
Data: A=(-4,+2,0)
```

```
      C=(-3,0,+1)
```

每个数组分别放在不同的页面中;而程序 Y 在运行过程中，其数组将依次用到程序空间的第 3,5,4,2,5,3,1,3,2,5,1,3,1,5,2 页。如果采用 LRU 算法，实存却只有 8 页位置可供存放数组之用。试问为这两首程序的数组分别分配多少个实页最为合适？为什么？

解答：

分别分配给程序 X 和 Y 的数组 4 个实页最为合适。

根据题意，程序 X 依次调用数组 A,C,B,B,E, A,C,B,B,C,A,D,D,E, A,C,B,B,E 中的数据。

设程序 X 中的数组 A,B,C,D,E 分别存放于程序空间的第 1,2,3,4,5 页，则程序的页地址流为：1, 3, 2, 2, 5, 1, 3, 2, 2, 3, 1, 4, 4, 5, 1, 3, 2, 2, 5。

分析使用 LRU 算法对程序 X 的页地址流进行堆栈处理的过程可知，分配给程序 X 的数组 5 个实页最为合适;分析使用 LRU 算法对程序 Y 的页地址流进行堆栈处理的过程可知，分配给程序 Y 的数组 4 个实页最为合适。

但实存只有 8 页位置可供存放数组之用，所以，分别分配给程序 X 和 Y 的数组 4 个实页。

note:

分时运行在微观上是串行的，就是说，分时运行时把时间划分为若干时间片，每个程序轮流占用时间片;在宏观上是并行的，就是说，每个程序在一个时间片内

并不能运行完。总的来看，是同时运行的，所以两个程序分配的实页和不能大于 8。

参考：上面的 FORTRAN 源代码转成 C 后

```
main()
{
int A[]={-4,2,0};
int C[]={-3,0,1};
for (i=0,i<3,i++)
{B[i]=A[i]-C[i];
if (B[i]<0)
E[i]=0;
else
{D[i]=2*C[i]-A[i];
if (D[i]<>0)
E[i]=0;
};
};
}
```

8. 设一个按位编址的虚拟存储器，它应可对应 1K 个任务，但在一段较长时间内，一般只有 4 个任务在使用，故用容量为 4 行的相联寄存器组硬件来缩短被变换的虚地址中的用户位位数；每个任务的程序空间最大可达 4096 页，每页为 512 个字节，实主存容量为 2^{20} 位；设快表用按地址访问存储器构成，行数为 32，快表的地址是经散列形成；为减少散列冲突，配有两套独立相等比较电路。请设计该地址变换机构，内容包括：

- (1) 画出其虚、实地址经快表变换之逻辑结构示意图；
- (2) 相联寄存器组中每个寄存器的相联比较位数；
- (3) 相联寄存器组中每个寄存器的总位数；
- (4) 散列变换硬件的输入位数和输出位数；

(5)每个相等比较器的位数;

(6)快表的总容量(以位为单位)。

解:

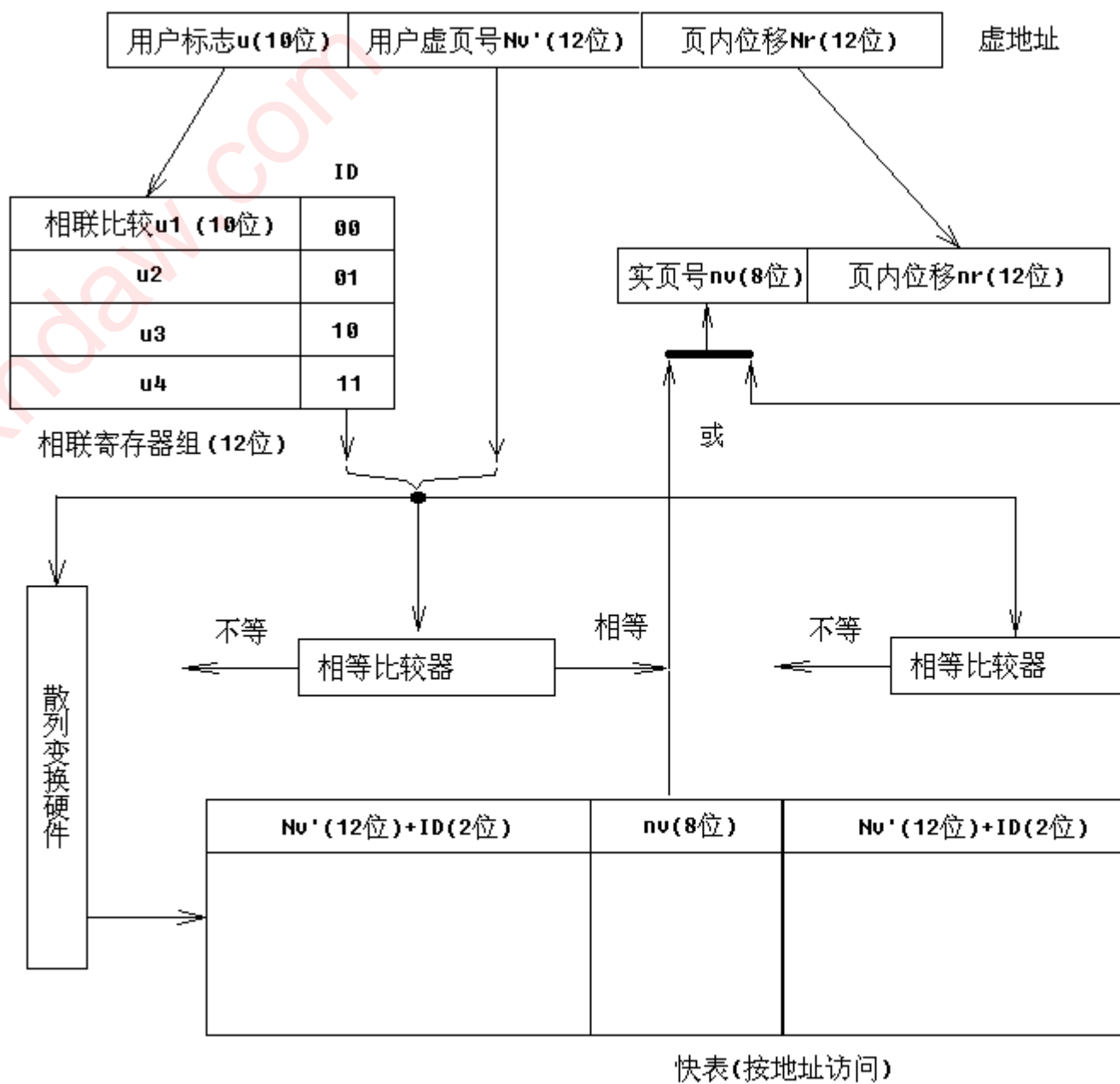
(1)依题意得知:

虚地址为 34 位, 其中用户号为 10 位(对应 1K 的任务)、虚页号 12 位(每个任务 4096 页)、页内位移 12 位(每页 512 字节, $512 \text{ 字节} = 512 * 8 = 1024 * 4 = 2^{12}$)

实地址为 20 位, 其中实页号 8 位, 页内位移 12 位(与虚页页内位移对应)

相联寄存器的作用: 把 10 位的用户号转换为 2 位的 ID(因为一般只有 4 个任务在使用), 并把 ID 与虚地址的虚页号合并到快表中查实页号。

快表的作用: 相当于页表, 即虚页号对实页号的对应关系。但又有所简化(原因是如果用用户号和虚页号与实页号对应, 前者就有 22 位, 现改进后虚页号只有 14 位了)



虚拟存储器快表示意图

- (2)相联寄存器组中每个寄存器的相联比较位数为 10（与虚地址中的用户号宽度对应）
- (3)相联寄存器组中每个寄存器的总数为 12（用户号宽度+ID 宽度）
- (4)散列变换硬件的输入位数为 14 位（虚页号宽度+相联寄存器中 ID 的宽度），输出位数为 8 位（与主存中的实页号宽度对应）

(5)每个相等比较器的位数=ID+用户虚页号 $nv'=2+12=14$ (位)。

(6)快表的总容量: $32 \text{ 行} \times (14(\text{输入位数})+8(\text{输出位数})) \times 2=32 \times 22 \times 2$

9.考虑一个 920 个字的程序,其访问虚存的地址流为 20, 22, 208, 214, 146, 618, 370, 490, 492, 868, 916, 728。

(1)若页面大小为 200 字,主存容量为 400 字,采用 FIFO 替换算法,请按访存的各个时刻,写出其虚页地址流,计算主存的命中率;

(2)若页面大小为 100 字,再做一遍;

(3)若页面大小为 400 字,再做一遍;

(4)由(1)、(2)、(3)的结果可得出什么结论?

(5)若把主存容量增加到 800 字,按第(1)小题再做一遍,又可得出什么结论?

解:

(1)主存容量 400 字,页面大小 200 字,所以主存实页数为 2;

把地址流转换为页地址流,以第一个虚地址流转换为页地址流为例说明:求模公式为: $INT(\text{地址}/\text{页面大小})$,就是把地址整除于页面大小,得 $INT(20/200)=0$,下同,所以页地址流为: 0,0,1,1,0,3,1,2,2,4,4,3

按 FIFO 算法得出替换过程为: 0(调入), 0(命中), 1(调入), 1(命中), 0(命中), 3(替换 0, 0 比 1 先入队, 所以被替换, 下同), 1(命中), 2(替换 1), 2(命中), 4(替换 3), 4(命中), 3(替换 2), 所以总共命中 6 次。

故命中率 $H=6/12=50\%$

(2)方法同(1) $H=25\%$

(3) $H=50\%$

(4)由以上结论可得, FIFO 算法的条件下,当页面大小发生变化时,其命中率变化是:一开始随页面大小增大命中率(第一步与第二步比较),但当页面大小增加到一定时,命中率不再增加(第一步与第三步比较)。

(5)命中率为 58%,结论是如果分配给主存容量增加时可以搞高命中率。

10. 在一个页式二级虚拟存储器中,采用 FIFO 算法进行页面替换,发现命中率 H 太低,因此有下列建议:

(1)增大辅存容量;

- (2)增大主存容量(页数);
- (3)FIFO 改为 LRU;
- (4)FIFO 改为 LRU, 并增大主存容量(页数);
- (5)FIFO 改为 LRU, 并增大页面大小。

试分析上述各建议对命中率的影响情况。

解答:

- (1)增大辅存容量, 对命中率 H 无影响。
- (2)增大主存容量(页数), 可普遍提高命中率。
- (3)FIFO 改为 LRU, 一般可提高命中率。
- (4)FIFO 改为 LRU, 并增大主存容量(页数), 一般可使命中率有较大提高。
- (5)FIFO 改为 LRU, 并增大页面大小, 如果原来页面很小, 则会使命中率显著上升, 如果原来页面很大, 则会使命中率下降。

11.采用组相联映象的 Cache 存储器, Cache 为 1KB, 要求 Cache 的每一块在一个主存周期内能从主存取得。主存模 4 交叉, 每个分体宽为 32 位, 总容量为 256KB。用按地址访问存储器构成相联目录表实现主存地址到 Cache 地址的变换, 并约定用 4 个外相等比较电路。请设计此相联目录表, 求出该表之行数、总位数及每个比较电路的位数。

解答:

设 Cache 地址中的组内块号为 s , 相联目录表的行数是 $2^{(13-s)}$, 总位数是 $(8+2s)*2^{(15-s)}$, 每个比较电路的位数为 $8+s$ 。

剖析:

在一个主存周期内主存能访问到的字节数为 $mW=4*32/8=16(\text{Byte})$ 。要求 Cache 的每一块在一个主存周期内能从主存取得, 所以, Cache 中每块的块内字数不能大于 16Bytes。为了加速调块, 一般让每块的大小等于在一个主存周期内主存能访问到的字数, 即 16Bytes。

设 Cache 地址中的组内块号为 s , 相联目录表的行数=Cache 地址内的组数 $Q=\text{Cache 容量}/(\text{每组块数}*\text{每块大小})=1\text{KB}/(2^4*32)=2^{13}/(2^s*2^7)=2^{(6-s)}$ 。

主存块数/Cache 块数=256=2*8, 所以, 主存地址中的区号 $nd=8$ 。每个比较电路的位数= $nd+s'=nd+s=8+s$ 。

相联目录表的总位数=表中子目录表的个数*每个子目录表的位数*相联目录表的行数 $=4*(nd+s'+s)*Q=4*(8+2s)*2^{(6-s)}=(8+2s)*2^{(8-s)}$ 。

note:

若认为相等比较电路的个数=组内块数，则相联目录表的行数 $=2^4$ ，每个比较电路的位数=10，相联目录表的总位数 $=12*2^6$ 。

12.有一个 Cache 存储器。主存共分 8 个块(0~7),Cache 为 4 个块(0~3),采用组相联映象，组内块数为 2 块，替换算法为近期最少使用算法(LRU)。

(1)画出主存、Cache 地址的各字段对应关系(标出位数)图;

(2)画出主存、Cache 空间块的映象对应关系示意图;

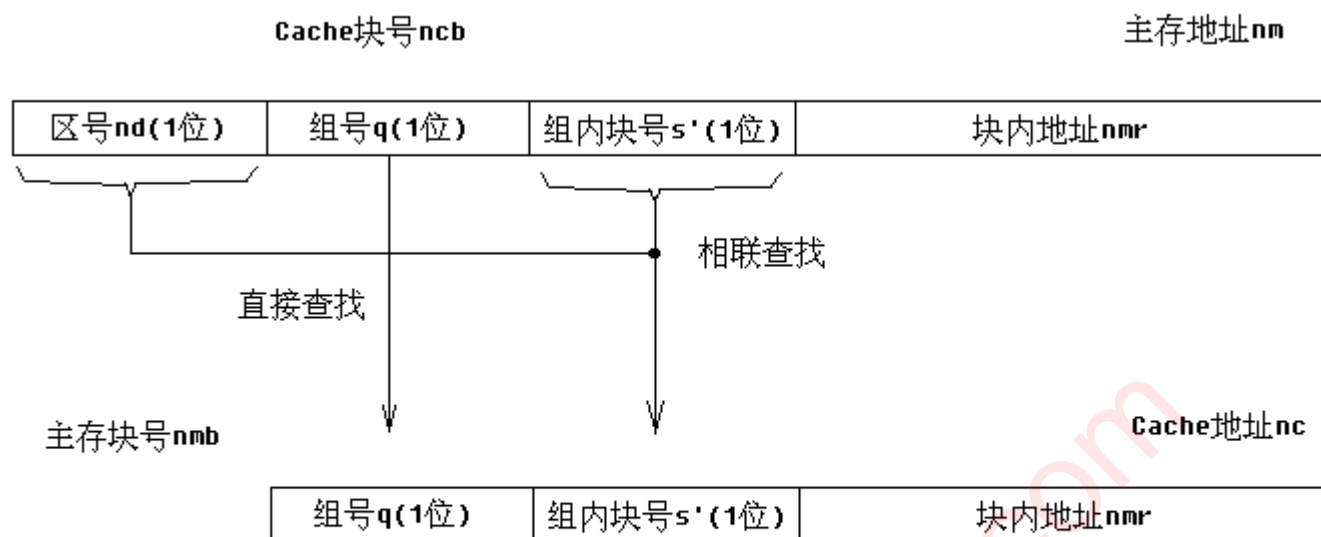
(3)对于如下主存块地址流：1,2,4,1,3,7,0,1,2,5,4,6,4,7,2,如主存中内容一开始未装入 Cache 中，请列出 Cache 中各块随时间的使用状况;

(4)对于(3),指出块失效又发生块争用的时刻;

(5)对于(3),求出此期间 Cache 的命中率。

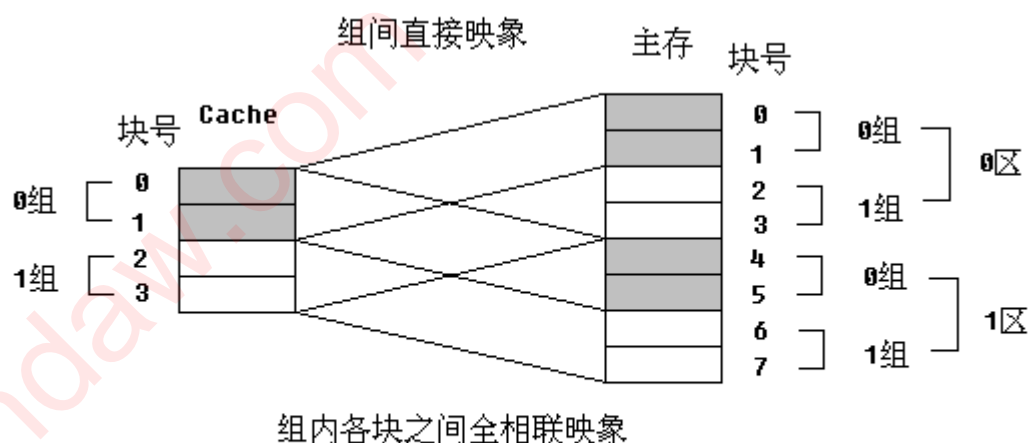
解答:

(1)主存地址、Cache 地址的各字段的位数及其对应关系如下图所示



主存地址和Cache地址的对应关系

(2)主存块、Cache 块的映象对应关系如下图所示



主存块和Cache块的对应关系

(3)Cache 中各块随时间的使用状况如下图所示。图中标*号的是候选替换块的块号，H：命中;R：替换;L：失效。

time t	1	2	3	4	5	6	7	8	9	10	11	12	13
主存块地址流	1	2	4	1	3	7	0	1	2	5	4	6	4
Cache块	0	1	1*	1	1	1	1*	1	1	1*	4	4	4
	1		4	4*	4*	4*	0	0*	0*	5	5*	5*	5*
	2		2	2	2*	7	7	7	7*	7*	7*	6	6
	3				3	3*	3*	3*	2	2	2	2*	2*
命中情况	L	L	L	H	L	R	R	H	R	R	R	R	H

Cache的使用情况

(4)发生块失效又发生块争用的时刻有 6、7、9、10、11、12、14、15。

(5)Cache 的块命中率 $H_c = 3/15 = 0.2$ 。

剖析：

由于主存块、Cache 块之间存在上述的映象对应关系，主存的第 0、1、4、5 块只能映象装入或替换物理 Cache 的第 0、1 块；主存的第 2、3、6、7 块只能映象装入或替换物理 Cache 的第 2、3 块。

13. 采用组相联映象，LRU 替换算法的 Cache 存储器，发现等效访问速度不高，为此建议：

- (1) 增大主存容量；
- (2) 增大 Cache 的块数(块的大小不变)；
- (3) 增大组相联组的大小(块的大小不变)；
- (4) 增大块的大小(组的大小和 Cache 总容量不变)；
- (5) 提高 Cache 本身器件的访问速度。

解答：

- (1) 增大主存容量对 Cache 的访问时间 t_a 基本不影响，从而对 Cache 的等效访问速度基本不影响。
- (2) 增大 Cache 的块数(块的大小不变)一般将使 Cache 的命中率 H_c 上升，从而使 t_a 下降，从而提高 Cache 的等效访问速度。
- (3) 增大组相联组的大小(块的大小不变)一般将使 Cache 的命中率 H_c 上升，从而使 t_a 下降，从而提高 Cache 的等效访问速度。
- (4) 增大块的大小(组的大小和 Cache 总容量不变)一般将使 t_a 下降，从而提高 Cache 的等效访问速度。
- (5) 提高 Cache 本身器件的访问速度一般将缩短 t_a ，从而提高 Cache 的等效访问速度。

14. 你对 Cache 存储器的速度不满，于是申请到一批有限的经费，为能发挥其最大经济效益，有人建议你再买一些同样速度的 Cache 片子以扩充其容量；而另有人建议你干脆去买更高速的 Cache 片子将现有的低速 Cache 片子全部换掉。你认为哪种建议可取？你如何做决定？为什么？

解答：

Cache 本身的速度与容量都会影响 Cache 存储器的等效访问速度。如果对 Cache 存储器的等效访问速度不满，需要改进的话，就要作具体分析，看看现在 Cache 存储器的等效访问速度是否已接近于 Cache 本身的速度。如果差得较远，

说明 Cache 的命中率低，应从提高 Cache 命中率着手，包括调整组的大小、块的大小、替换算法以及增大 Cache 容量等。如果 Cache 存储器的等效访问速度已经非常接近于 Cache 本身的速度还不能满足需要，就应该更换更高速的 Cache 片子。

第五章课后题

1、假设解释分取指、分析与执行3步，每步的时间相应为 $t_{\text{取指}}$ 、 $t_{\text{分析}}$ 、 $t_{\text{执行}}$ ，

(2) 分别在 $t_{\text{取指}}=t_{\text{分析}}=2$ 、 $t_{\text{执行}}=1$ 及 $t_{\text{取指}}=t_{\text{执行}}=5$ 、 $t_{\text{分析}}=2$ 两种情况下，计算出下述各结果。

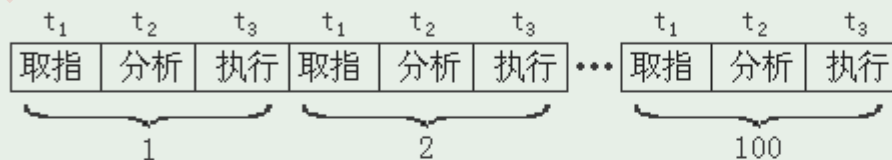
(1) 分别计算下列几种情况下，执行完100条指令所需时间的一般关系式：

(i) 顺序方式；

解： $t=(t_1+t_2+t_3)*100$

① $t_1=t_2=2$ ， $t_3=1$ 时， $t=(2+2+1)*100=500$

② $t_1=t_3=5$ ， $t_2=2$ 时， $t=(5+2+5)*100=1200$

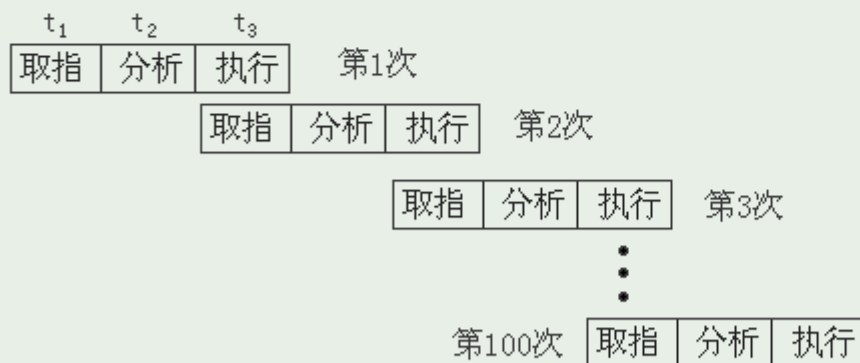


(ii) 仅“执行_k”与“取指_{k+1}”重叠

解： $t=t_1+100*t_2+99*\max(t_1, t_3)+t_3$

① $t_1=t_2=2$ ， $t_3=1$ 时， $t=2+100*2+99*2+1=401$

② $t_1=t_3=5$ ， $t_2=2$ 时， $t=5+100*2+99*5+5=705$

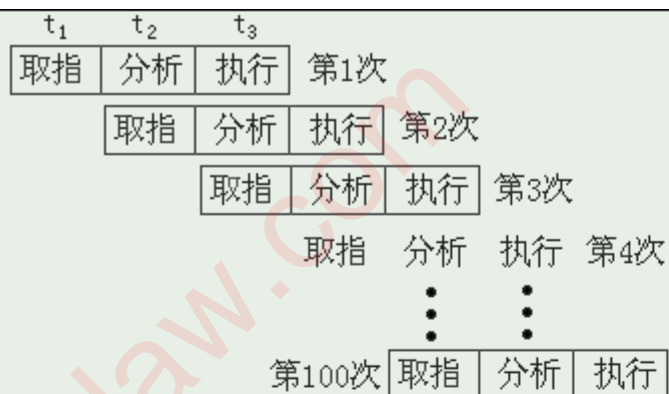


(iii) 仅“执行_k”、“分析_{k+1}”、“取指_{k+2}”重叠。

解： $t=t_1+\max(t_1, t_2)+98*\max(t_1, t_2, t_3)+\max(t_2, t_3)+t_3$

① $t_1=t_2=2$ ， $t_3=1$ 时， $t=2+2+98*2+2+1=203$

② $t_1=t_3=5$ ， $t_2=2$ 时， $t=5+5+98*5+5+5=510$

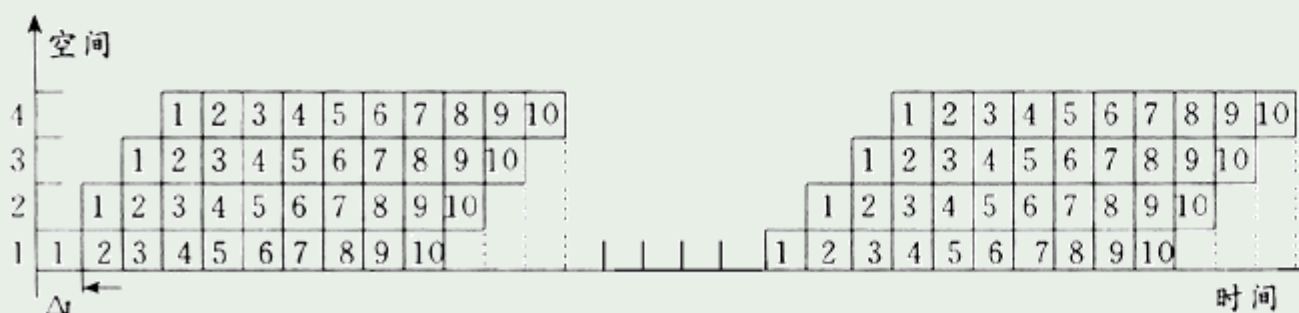


2、流水线由4个功能部件组成，每个功能部件的延迟时间为 Δt ，当输入10个数据后间歇 $5\Delta t$ ，又流水线的吞吐率，并画出其时空图。

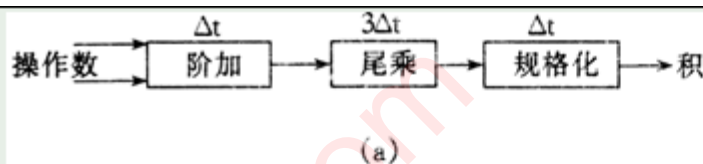
解：设流水线共处理K组数据（每组10个数据）

$$\text{则流水线吞吐率} = \frac{10K}{13K\Delta t + 5\Delta t (K-1)} = \frac{10K}{18K\Delta t - 5\Delta t}$$

时空图如下：



3、有一个浮点乘流水线如图5.35(a)所示，其乘积可直接返回输入端或暂存于相应缓冲寄存器中，并求出该流水线吞吐率和效率；当流水线改为图5.35(b)形式实现同一计算时，求该流水线

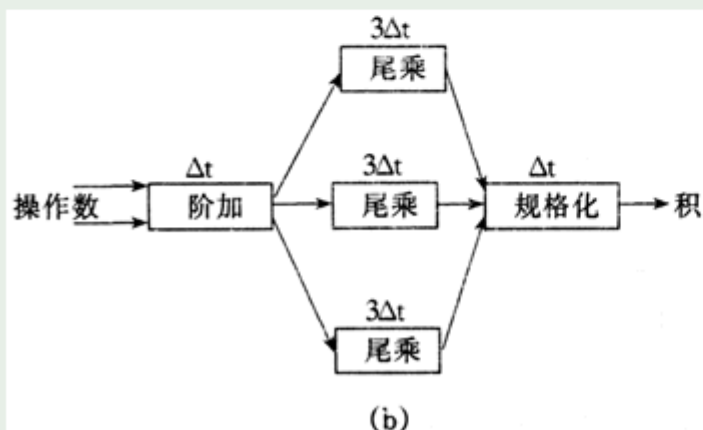
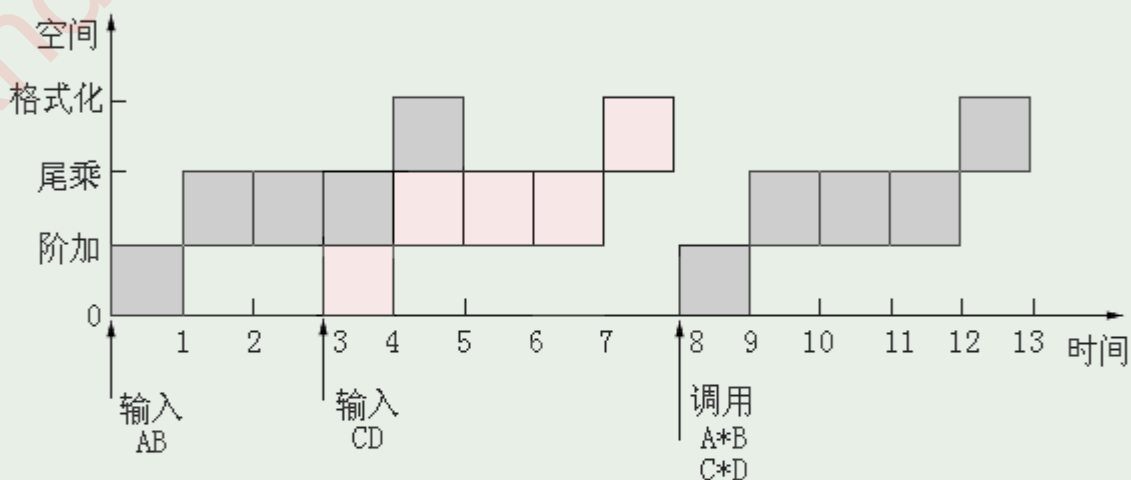


(1) 吞吐率 $= 3/13\Delta t$

效率 $= 15 * \Delta t / (3 * 13\Delta t) = 38.5\%$

P.S. 吞吐率 = 任务总数 / 时间和

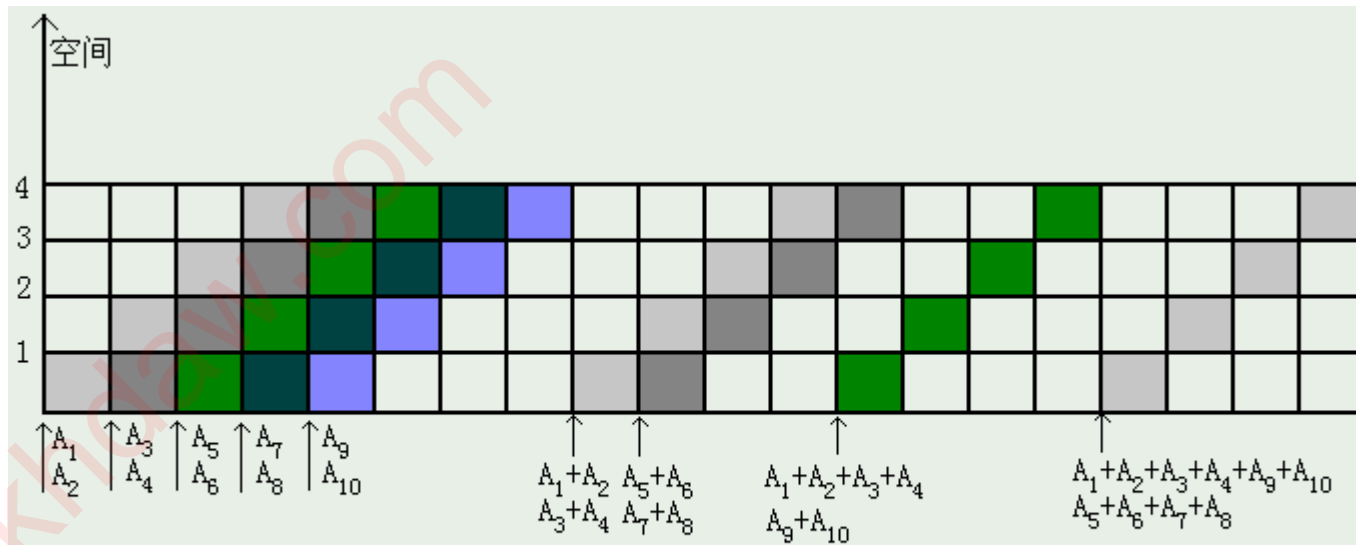
效率 = 实用小块数 / 矩形面积



(2) 吞吐率 $= 5/11\Delta t$

效率 $= 15 * \Delta t / (5 * 11\Delta t) = 27.3\%$

4、一个4段的双输入端规格化浮点加法流水线，每段经过时间10ns,输出可直接返回输入或将结果暂存于相应缓冲器中，问最少需经多少时间能求 $(10) \sum_{i=1}^n A_i$ ，并画出时空图。



吞吐率 = $9 / (20 * 10\text{ns}) = 4.5 * 10^{10}$

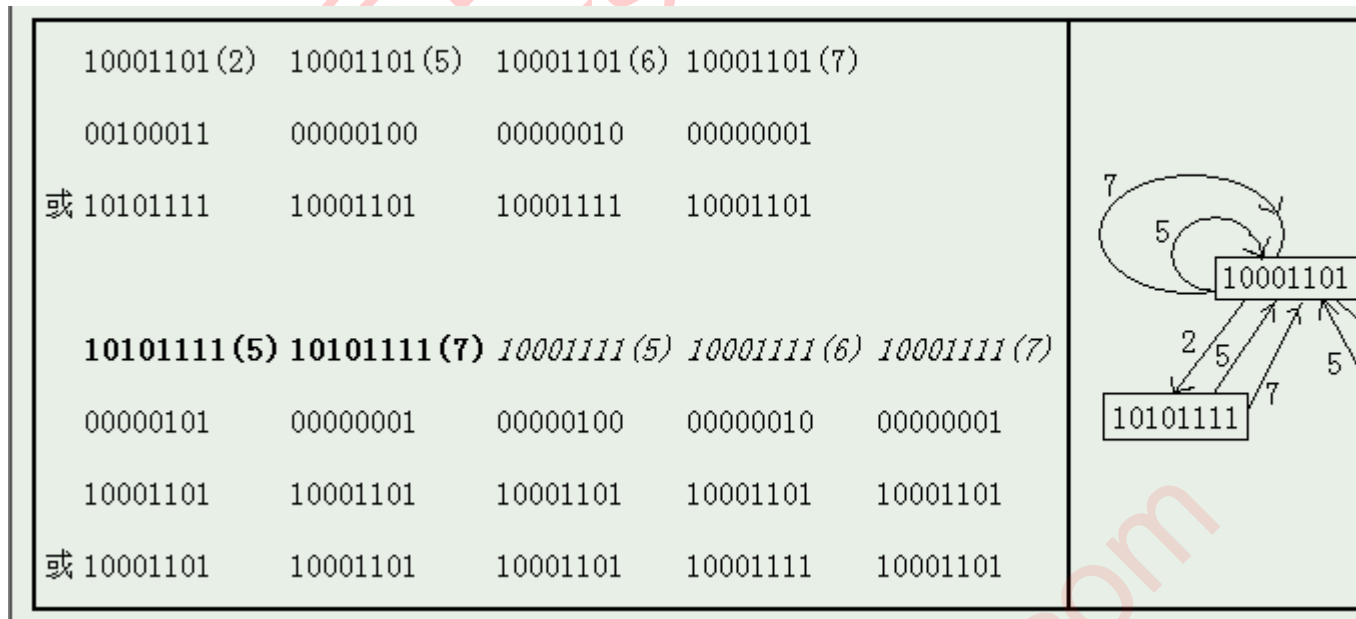
效率 = $(20 + 8 + 4 + 4) * \Delta t / (20 * 4 * \Delta t) = 45\%$

9、

解：根据预约表，延迟禁止表 $F = \{1, 3, 4, 8\}$

$C = 10001101$

由于初始冲突向量的 c_2, c_5, c_6, c_7 为 0，所以第二个任务可以距第一个任务 2, 5, 6 或 7 拍流入流水线。

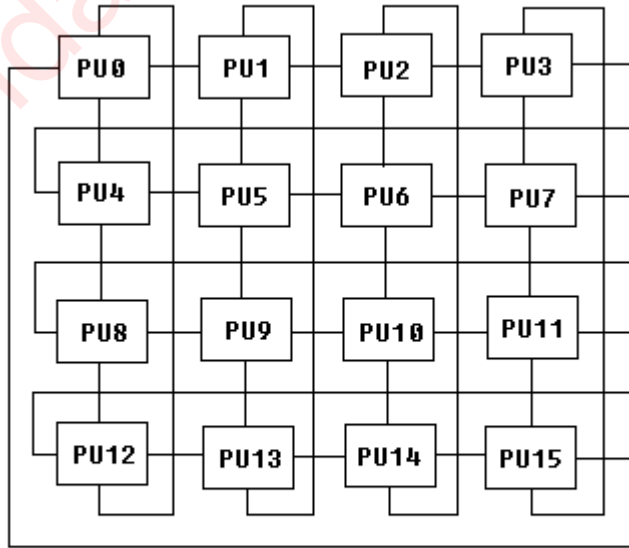


第六章课后题

1.画出 16 台处理器仿 ILLIAC IV 的模式进行互连的互连结构图，列出 PE0 分别只经一步、二步和三步传送能将信息传送到的各处理器号。

答：

6 台处理器仿 ILLIAC IV 处理单元的互连结构如图所示：



仿ILLIAC IV 处理单元的互连结构

图中第个 PU 中包含 PE、PEM 和 MLU。

PE0(PU0)经一步可将信息传送至 PU1、PU4、PU12、PU15。

PE0(PU0)至少需经二步才能将信息传送至 PU2、PU3、PU5、PU8、PU11、PU13、PU14。

PE0(PU0)至少需经三步才能将信息传送至 PU6、PU7、PU9、PU10。

2.编号为 0、1、...、15 的 16 个处理器，用单级互连网互连。当互连函数分别为

- (1)Cube3
- (2)PM2+3
- (3)PM2-0
- (4)Shuffle
- (5)Shuffle(Shuffle)

时，第 13 号处理器各连至哪一个处理器？

解答：

(1)5 号处理器

(2)5 号处理器

(3)12 号处理器

(4)11 号处理器

(5)7 号处理器

剖析：

由题意知，有 16 个处理器，即 $N=16, n=\log_2(N)=\log_2(16)=4$ 。

$\text{Cube}_3(13)=\text{Cube}_3(1101)=0101=5$

$\text{PM}_{2+3}(13)=(13+2^3)\bmod 16=5$

$\text{PM}_{2-0}(13)=(13-2^0)\bmod 16=12$

$\text{Shuffle}(13)=\text{Shuffle}(1101)=1011=11$

$\text{Shuffle}(\text{Shuffle})=\text{Shuffle}(11)=\text{Shuffle}(1011)=0111=7$

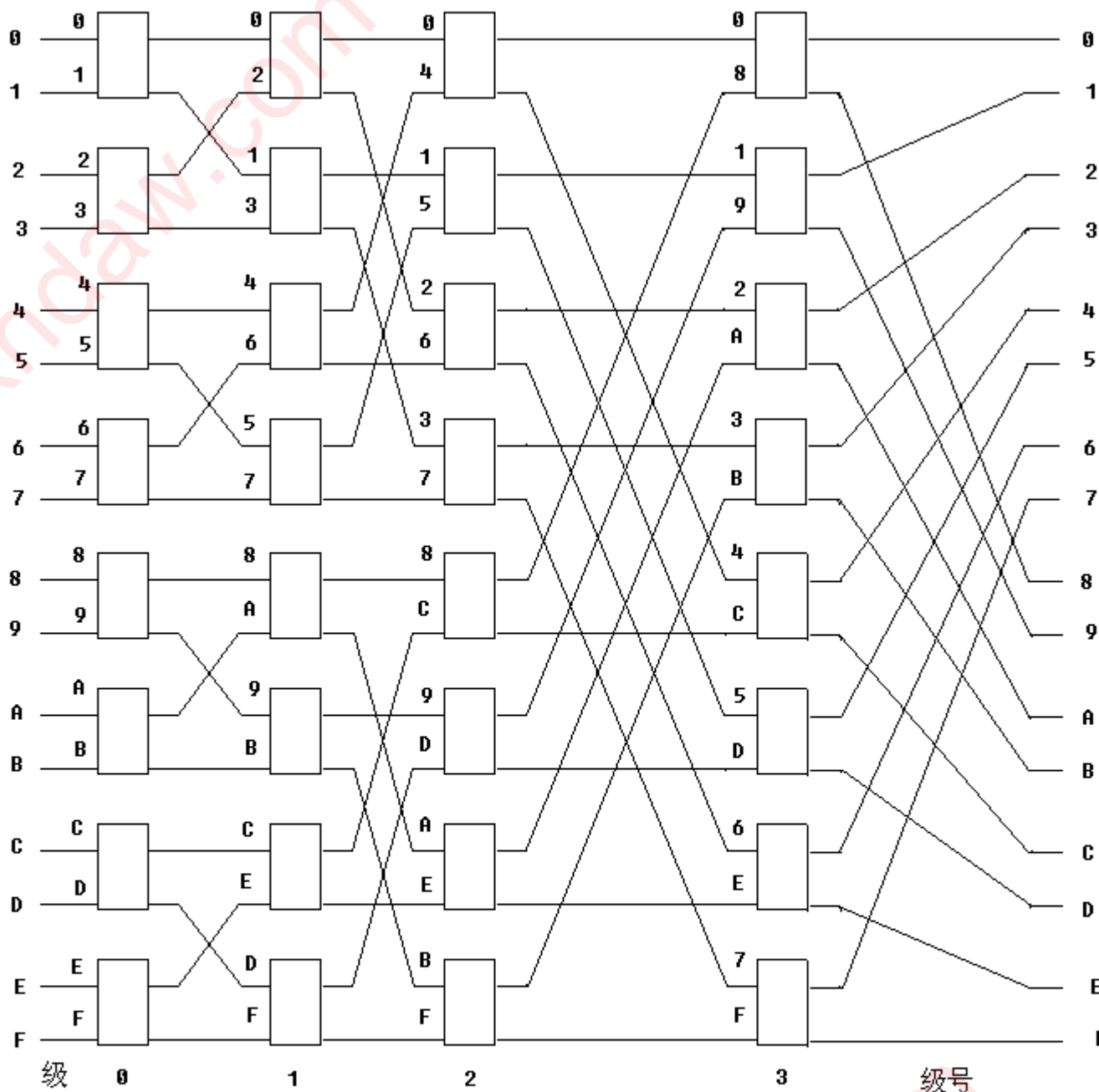
3.编号分别为 0、1、2、...、F 的 16 个处理器之间要求按下列配对通信：(B、1)，(8、2)，(7、D)，(6、C)，(E、4)，(A、0)，(9、3)，(5、F)。试选择所用互连网络类型、控制方式，并画出该互连网络的拓补结构和各级交换开关状态图。

解答：

采用 4 级立方体网络，级控制。该互连网络的拓补结构和各级交换开关状态图如下图所示：

入端

出端



级

Cube0

Cube1

Cube2

Cube3

级号

开关状态 直连

交换

直连

交换

多级互连网络

剖析:

从处理器号的配对传送关系可以转成处理器二进制编号的配对传送关系：

(B,1) (1011,0001)

(8,2) (1000,0010)

(7,D) (0111,1101)

(6,C) (0110,1100)

(E,4) (1110,0100)

(A,0) (1010,0000)

(9,3) (1001,0011)

(5,F) (0101,1111)

不难得出其一般规律是：二进制编号为 $P_3P_2P_1P_0$ 的处理器与 $(\neg P_3)P_2(\neg P_1)P_0$ 的处理器配对交换数据。由于实现的都是交换函数的功能，采用成本最低的级控制多级立方体互联网络就可以实现。

$N=16$ 的多级立方体网络，由 $n=\log_2(16)=4$ 组成。每一级均使用 $N/2=8$ 个二功能交换开关。多级网络各级的级号由入端到出端依次为 0、1、2、3。第 i 级的每个交换单元的配对用的是 $Cube_i(P_3...P_i...P_0)=P_3...(\neg P_i)...P_0$ 函数。根据本题的要求，应当让第 1、3 级的各交换单元处于“交换”状态，第 0、2 级的各交换单元处于“直连”状态。

4.画出编号为 0、1、...、F 共 16 个处理器之间实现多级立方体互连的互连网络，采用级控制信号为 1100(从右至左分别控制第 0 级至第 3 级)时，9 号处理器连向哪个处理器？

解答：

多级立方体互连网络的图和第 3 题的图基本一致，不同之处在于，第 0、1 级的开关状态为直连，第 2、3 级的开关状态为交换。

9 号处理器在经过 0 级和 1 级交换开关后，连向第 10 个处理器；在经过 2 级交换开关后，连向第 4 个处理器；在经过 3 级交换开关后，连向第 9 个处理器。

5.对于采用级控制的三级立方体网络，当第 i 级($0 \leq i \leq 2$)为直连状态时，不能实现哪些结点之间的通信？为什么？反之，当第 i 级为交换状态呢？

解答：

当第 i 级为直连状态时，不能实现入、出两端的处理器二进制编码的编号中，第 P_i 位取反的处理器之间的连接。例如，第 0 级为直连状态时，入端号为 $\times \times 0$ 的处理器仅能与出端号为 $\times \times 0$ 的处理器进行数据传送，不能与出端号为 $\times \times 1$ 的处理器进行数据传送。因为交换开关的直连状态被定义为 i 入连 i 出， j 入连 j 出，所以，反映出实现互连的入、出端号的二进制码

中的 P_i 位是不能变反的，其它的各位可以不变，也可以变反。

当第 i 级为交换状态时，不能实现入、出两端的处理器二进制编码的编号中，第 P_i 位相同的处理器之间的连接。例如，第 0 级为交换状态时，入端号为 $\times\times 0$ 的处理器仅能与出端号为 $\times\times 1$ 的处理器进行数据传送，不能与出端号为 $\times\times 0$ 的处理器进行数据传送。因为交换开关的直连状态被定义为 i 入连 j 出， j 入连 i 出，所以，反映出实现互连的入、出端号的二进制码中的 P_i 位必须变反，其它的各位可以不变，也可以变反。

6. 假定 8×8 矩阵 $A=(a_{ij})$ ，顺序存放在存储器的 64 个单元中，用什么机关报单级互连网络可实现对该矩阵的转置变换？总共需要传送多少步？

解答：

采用单级混洗互连网络可实现对 8×8 矩阵的转置变换，共需传送 3 步。

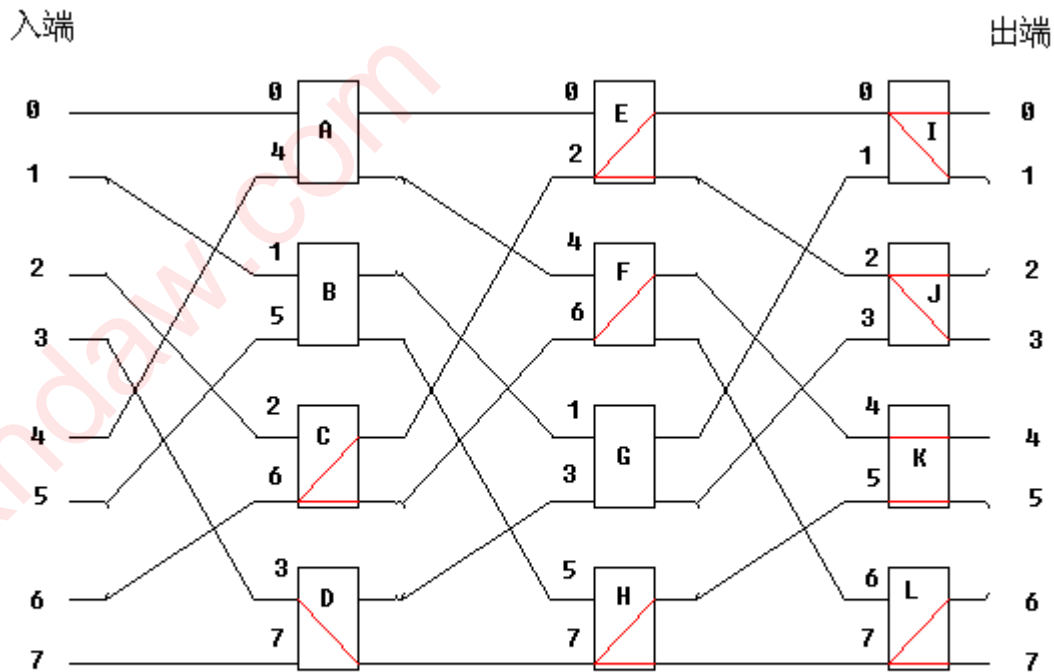
剖析：

8×8 矩阵中任一元素 a_{ij} ，它在存储器中所占的位置是 $i \cdot 8 + j$ (即 $i \cdot 2^3 + j$)。每个元素的行坐标和列坐标均用 3 位表示，设 $b_5b_4b_3$ 为行下标的二进制编号， $b_2b_1b_0$ 为列下标的二进制编号，经过 3 次全混洗后，元素下标号 $b_5b_4b_3b_2b_1b_0$ 就变成了 $b_2b_1b_0b_5b_4b_3$ ，即行下标的二进制编号改成了 $b_2b_1b_0$ ，列下标的二进制编号改成了 $b_5b_4b_3$ ，这样，就实现了矩阵的行列转置。

7. 画出 0~7 号共 8 个处理器的三级混洗交换网络，在该图上实现将 6 号处理器数据播送给 0~4 号，同时将 3 号处理器数据播送给其余 3 个处理器时的各有关交换开关的控制状态。

解答：

8 个处理器的三级混洗交换网络及其交换开关控制状态设置如下图所示：



多级混洗网络

8. 并行处理机有 16 个处理器要实现相当于先 4 组 4 元交换, 然后是 2 组 8 元交换, 再次是 1 组 16 元交换的交换函数功能, 请写出此时各处理器之间所实现的互连函数的一般式, 画出相应多级网络的拓扑结构图, 标出各组交换形状的状态。

解答:

互连函数的一般式为: $Cube_i(P_3P_2P_1P_0) = (\overline{P_3}P_2\overline{P_1}P_0)$ 。

多级立方体互连网络的拓扑结构图和第 3 题的图基本一致, 不同之处在于, 第 0、1、3 级的开关状态为直连, 第 2 级的开关状态为交换。

9. 具有 $N=2^n$ 个输入端的 Omega 网络, 采用单元控制。

(1) N 个输入总共可有多少种不同的排列;

(2) 该 Omega 网络通过一次可以实现的置换可有多少种是不同的;

(3) 若 $N=8$, 计算出一次通过能实现的置换数占全部排列数的百分比。

解答:

(1) N 个输入总共可有 $N!$ 种不同的排列。

(2) 该 Omega 网络通过一次可以实现的置换可有 $2^{((N/2)\log_2(N))} = N^{(N/2)}$ 种是不同的。

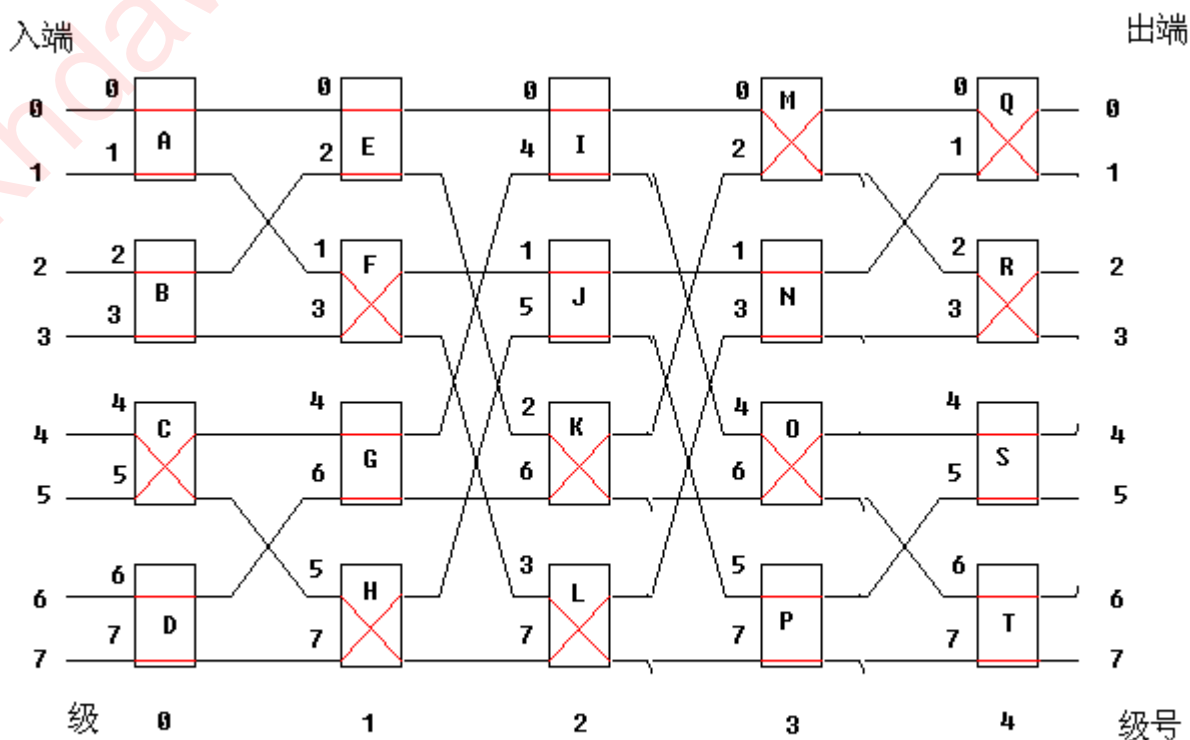
(3) 若 $N=8$, 通过 Omega 网络一次可以实现的不重复置换有 $8^4=4096$ 种; 8 个输入总共可实现的不重复排列有 $8!=40320$ 种。所以, 一次通过能实现的置换数占全部排列数的百分比为

$4096/40320 \times 100\% \approx 10.16\%$

10.画出 $N=8$ 的立方体全排列多级网络, 标出采用单元控制, 实现 $0 \rightarrow 3, 1 \rightarrow 7, 2 \rightarrow 4, 3 \rightarrow 0, 4 \rightarrow 2, 5 \rightarrow 6, 6 \rightarrow 1, 7 \rightarrow 5$ 的同时传送时的各交换开关的状态;说明为什么不会发生阻塞。

解答:

实现 $N=8$ 的立方体全排列多级网络及交换形状状态如下图所示



多级互连网络

在一到的映射时, 交换开关的状态组合有许多冗余, 所以不会发生阻塞。

11.在 16 台 PE 的并行处理机上, 要对存放在 M 个分体并行存储器中的 16×16 二维数组实现行、列、主对角线、次对角线上各元素均无冲突访问, 要求 M 至少为多少? 此时数组在存储器中应如何存放? 写出其一般规则。同时, 证明这样存放同时也可以无冲突访问该二维数组中任意 4×4 子阵的各元素。

解答:

M 至少为 17。

数组 A 中的任意一个元素 A_{ab} 的存放规则: 体号地址 $j = (4a + b) \bmod 17$, 体内地址 $i = a$, 按照对应关系将各数组元素存放到 $m=17$ 的并行存储器中, 如下图。由图可见, 这样存放同时也可以无冲突访问该二维数组中任意 4×4 子阵的各元素。

16×16 二维数组在并行存储器中存放的例子($m=17, n=16$)

体内地址 i	存储体号 j										
	0	1	2	3	4	5	6	7	8	9	10
0	a00	a01	a02	a03	a04	a05	a06	a07	a08	a09	a10
1	a113	a114	a115		a10	a11	a12	a13	a14	a15	a16
2	a29	a210	a211	a212	a213	a214	a215		a20	a21	a22
3	a35	a36	a37	a38	a39	a310	a311	a312	a313	a314	a315
4	a41	a42	a43	a44	a45	a46	a47	a48	a49	a410	a411
5	a514	a515		a50	a51	a52	a53	a54	a55	a56	a57
...

剖析：

设 16×16 的二维数组在并行存储器中同一列两个相邻元素地址错开的距离为 δ_1 ，同一行两个相邻元素地址错开的距离为 δ_2 ，当 m 取成 2^{2P+1} 时，实现无冲突访问的充分条件是 $\delta_1 = 2^P$ ， $\delta_2 = 1$ 。

对于这道题来说， $M=17=2^{(2 \times 2)+1}$ ，所以 $P=2$ 。 $\delta_1=2^P=4$ ， $\delta_2=1$ 。

数组存放的规则：体号地址 $j = (a \cdot \delta_1 + b \cdot \delta_2 + c) \bmod m$ (c 为 A_{00} 所在体号地址)， $i=a$ 。

第七章课后题

1.多处理机在结构、程序并行性、算法、进程同步、资源分配和调试上与并行处理机有什么差别?

答:

多处理机与并行处理机的主要差别是并行性的等级不同。

(1) 结构灵活性。多处理机制结构灵活性高于并行处理机。

(2) 程序并行性。多处理是指令、任务、作业并行,并行性的识别较难;并行处理机是操作级并行,并行性的识别较易。

(3) 并行任务派生。并行处理机工作能否并行工作由指令决定,多处理机必须有专门指令指明程序能否并行执行,派生的任务数是动态变化的。

(4) 进程同步。并行处理机的进程同步是自然的,而多处理机必须采取同步措施。

(5) 资源分配和任务调度。多处理机的资源分配和任务调度比并行处理机复杂得多。

2.多处理机有哪些基本特点?发展这种系统的主要目的可能有哪些?多处理着重解决哪些技术问题?

答:

○多处理机的基本特点:

多处理机具有两台以上的处理机,在操作系统控制下通过共享的主存或输入/输出子系统或高速通讯网络进行通讯.结构上多个处理机用多个指令部件分别控制,通过机间互连网络通讯;算法上不只限于处理向量数组,还要实现更多通用算法中的并行;系统管理上要更多地依靠软件手段,有效解决资源分配和管理,特别是任务分配,处理机调度,进程的同步和通讯等问题.

○使用多处理机的目的:

一是用多台处理进行多任务处理协同求解一个大而复杂的问题来提高速度,二是依靠冗余的处理机及其重组来提高系统的可靠性,适应性和可用性.

○多处理着重解决的技术问题:

(1) 硬件结构上,如何解决好处理机、存储器模块及 I/O 子系统间的互连。

(2) 如何最大限度开发系统的并行性,以实现多处理要各级的全面并行。

(3) 如何选择任务和子任务的大小，即任务的粒度，使并行度高，辅助开销小。

(4) 如何协调好多处理机中各并行执行任务和进程间的同步问题。

(5) 如何将任务分配到多处理机上，解决好处理机调度、任务调度、任务调度和资源分配，防止死锁。

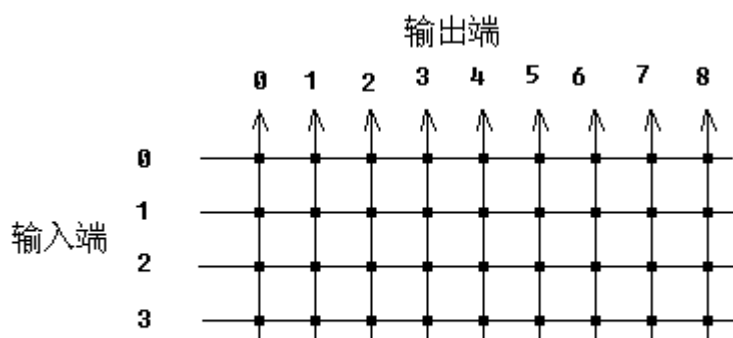
(6) 一旦某个处理发生故障，如何对系统进行重新组织，而不使其瘫痪。

(7) 多处理机机数增多后，如何能给编程者提供良好的编程环境，减轻程序的复杂性。

3. 分别画出 4×9 的一级交叉开关以及用两级 2×3 的交叉开关组成的 4×9 的 Delta 网络，比较一下交叉开关设备量的多少？

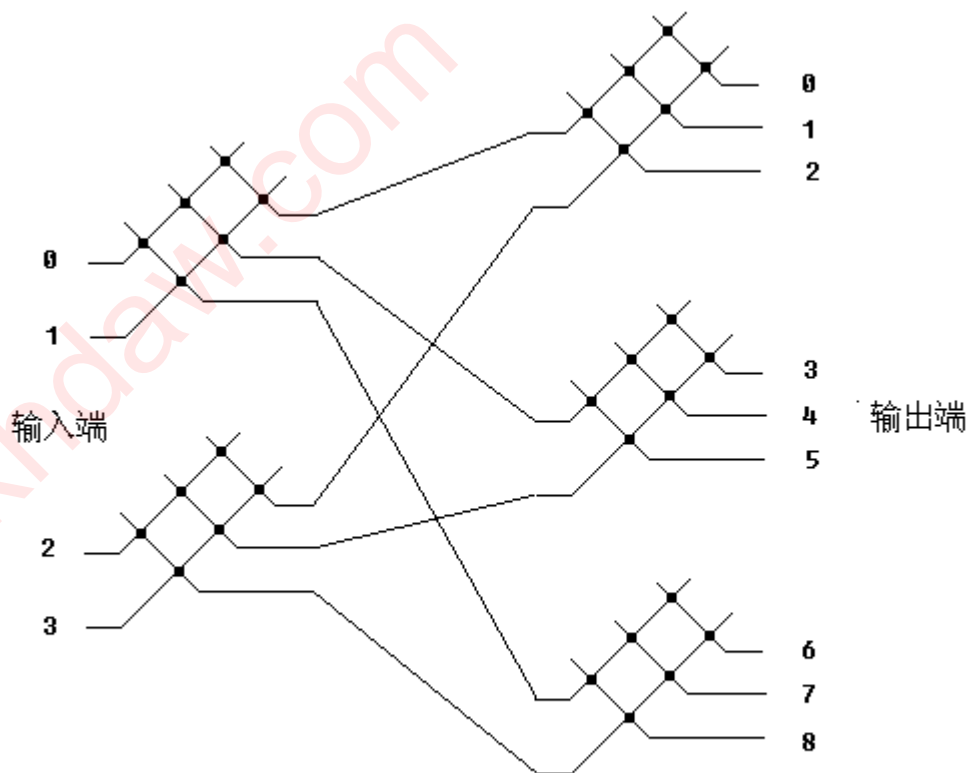
解答：

4×9 的一级交叉开关如下图所示：



一级 4×9 交叉开关

两级 2×3 的交叉开关组成的 4×9 的 Delta 网络如下图所示：



$2^2 \times 3^3$ 的 Delta 网络

$2^2 \times 3^3$ 有 Delta 网络由 5 个 2×3 的交叉开关组成，其交叉开关的结点数由一级网络的 36 个减少到现在 Delta 网络中的 $2 \times 3 \times 5 = 30$ 个。

剖析：

第一级有 2 个 2×3 的交叉开关，第 2 级有 3 个 2×3 的交叉开关，级间采用混洗拓扑。

4.说明 4×4 交叉开关组成的两级 16×16 交叉开关网络虽节省了设备，但它是一个阻塞式网络。

答：

16×16 交叉开关网络需要 256 个开关结点，每个结点中选 1 的多路裁决和选择电路。采用 4×4 的交叉开关构成的二级交叉开关网络，共需要 $16 \times 8 = 128$ 个开关结点，每个结点只需要 4 中选 1 的多路裁决和选择电路，节省了设备。但它是一个阻塞式网络。因为第 1 级每 4 个输入端中只能有一个连到第 2 级的一个输入端，而第 2 级的这个输入端本可以对应 4 个输出端的某一个。这就意味着，当第 1 级 4 个输入端中的某一个连到了最终的某个输出端时，第 1 级同组内其

它 3 个输入端由于有路径冲突，就不能同时将信息传送到第 2 级输出相应的另外 3 个输出端上，而采用 16×16 的一级交叉开关就不存在这种问题。

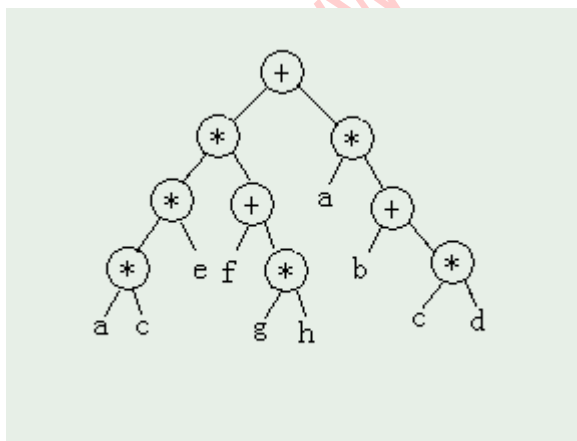
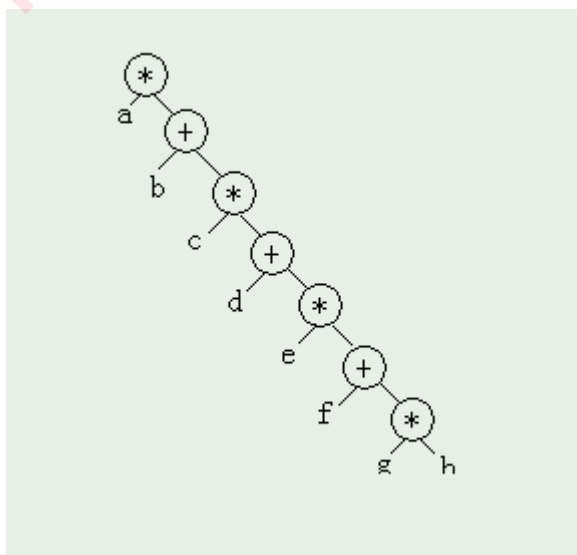
5. 由霍纳法则给定的表达式如下: $E = a(b + c(d + e(f + gh)))$ 利用减少树高的办法来加速运算，要求

(1) 画出树形流程图;

(2) 确定 T_p 、 P 、 S_p 、 E_p 诸值。

解答:

(1) 对于原式，单处理机串行运算树形流程图如左下图所示，多处理机并行运算树形流程图如右下图所示。



(2) P 台处理机运算的级数 $T_p = 4$ 。

所需处理机数目 $P = 3$ 。

加速比 $S_p = \text{顺序运算的级数 } T_1 / P$ 台处理机运算的级数 $T_p = 7/4$ 。

效率 $E_p = \text{加速比 } S_p / \text{所需处理机数目 } P = 7/12$ 。

6、求 A_1 、 A_2 ... A_8 的累加和，有如下程序：

S1 $A_1=A_1+A_2$

S2 $A_3=A_3+A_4$

S3 $A_5=A_5+A_6$

S4 $A_7=A_7+A_8$

S5 $A_1=A_1+A_3$

S6 $A_5=A_5+A_7$

S7 $A_1=A_1+A_5$

(1) 写出用 **FORK**、**JOIN** 语句表示其并行任务的派生和汇合关系的程序，以假想使此程序能在多处理机上运行。

(2) 画出该程序在有 3 台处理机制系统上运行的时间关系示意图。

(3) 画出该程序在有 2 台处理机制系统上运行的时间关系示意图。

解答：

(1)用 **FORK**、**JOIN** 语句表示其并行任务的派生和汇合关系的程序如下。

FORK 20

FORK 30

FORK 40

10 $A_1=A_1+A_2$

JOIN 4

GOTO 80

20 $A_3=A_3+A_4$

JOIN 4

GOTO 80

30 $A_5=A_5+A_6$

JOIN 4

GOTO 80

40 $A_7=A_7+A_8$

JOIN 4

80 **FORK** 60

50 A1=A1+A3

JOIN 2

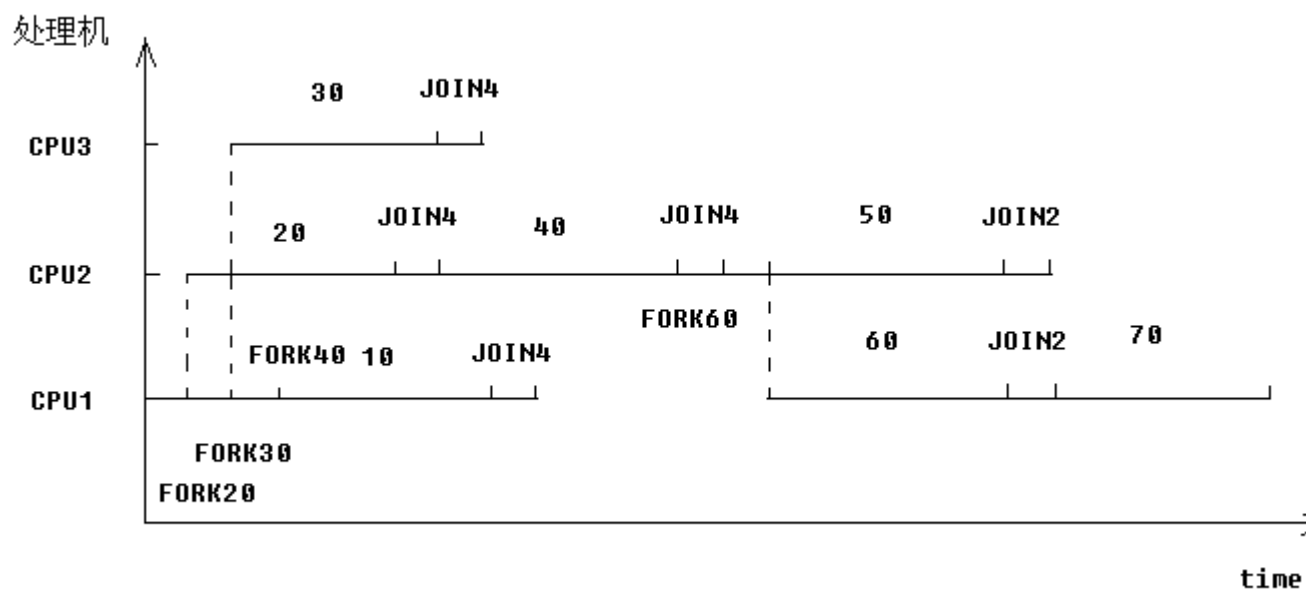
GOTO 70

60 A5=A5+A7

JOIN 2

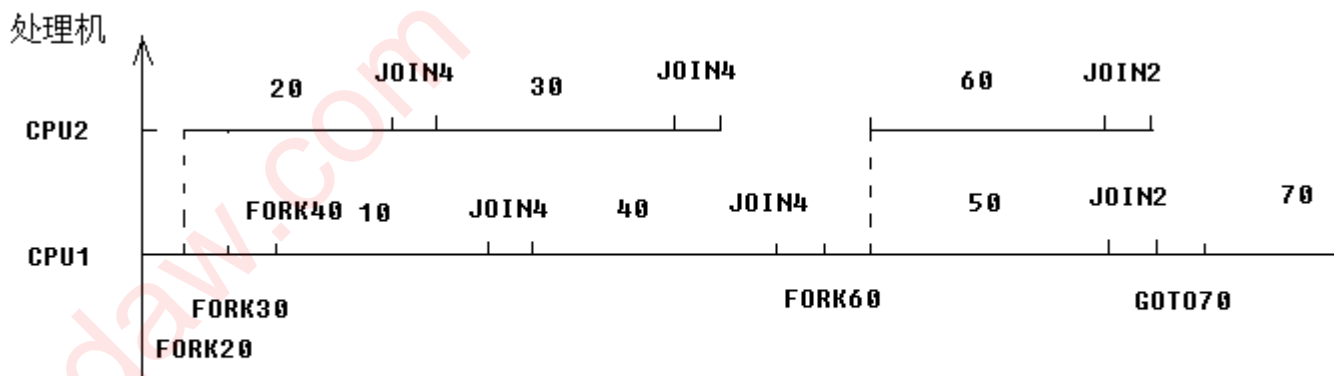
70 A1=A1+A5

(2)在有 3 台处理机制多处理机系统上运行的资源时间图如下图所示。假设标号为 50 和 60 的两个并发进程中，标号为 60 的进程最后完成。



程序在多处理机上运行的资源时间图

(3)在有 2 台处理机制多处理机系统上运行的资源时间图如下图所示。假设标号为 50 和 60 的两个并发进程中，标号为 50 的进程最后完成。



程序在多处理机上运行的资源时间图

剖析：

GOTO 70 语句的问题关键是 70 语句是在 50 语句还是 60 语句所在 CPU 上执行的。也就是说 50 语句和 60 语句谁先执行完。

7、若有如下程序：

$V=U/B$

$W=A*U$

$X=W-V$

$Y=W*U$

$Z=X/Y$

试用 FORK、JOIN 语句改写成可在多处理机上并行执行的程序。假设现有两台处理机，且除法速度最慢，加、减法速度最快，请画出该程序运行时的资源时间图。

解答：

用 FORK、JOIN 语句改写成可在多处理机上并行执行的程序如下：

S1 $U=A+B$

FORK S3

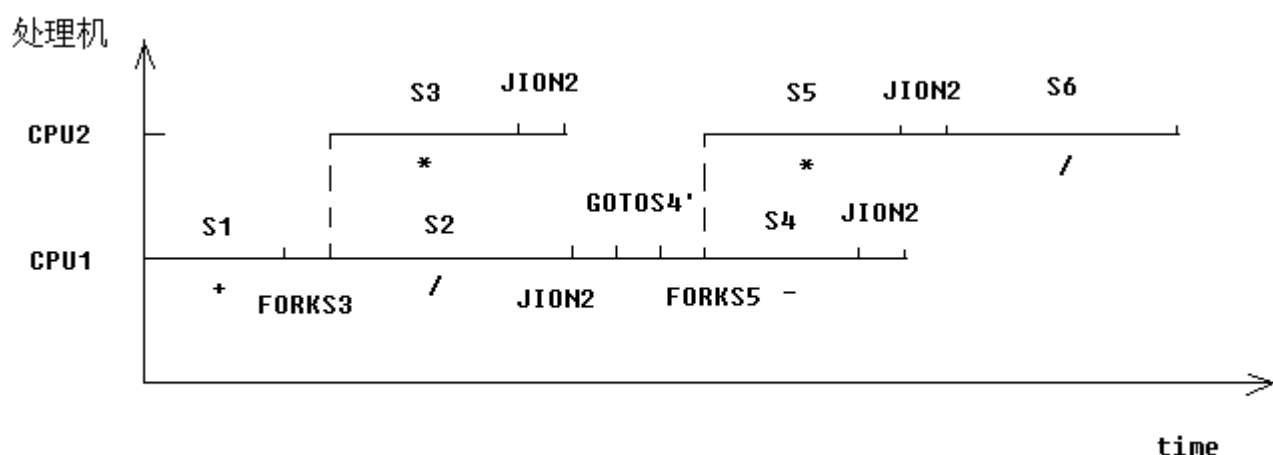
S2 $V=U/B$


```

JOIN 2
GOTO S4'
S3 W=A*U
JOIN 2
S4' FORK S5
S4 X=W-V
JOIN 2
GOTO S6
S5 Y=W*U
JOIN 2
S6 Z=X/Y

```

该程序在有 2 台处理机的多处理机系统上运行时的资源时间图如下所示：



程序在多处理机上运行的资源时间图

8. 分别确定下列各计算机系统中，计算点积 $S = (8) \sum_{i=1}^8 a_i * b_i$ 所需的时间(尽可能给出时空图示意)：

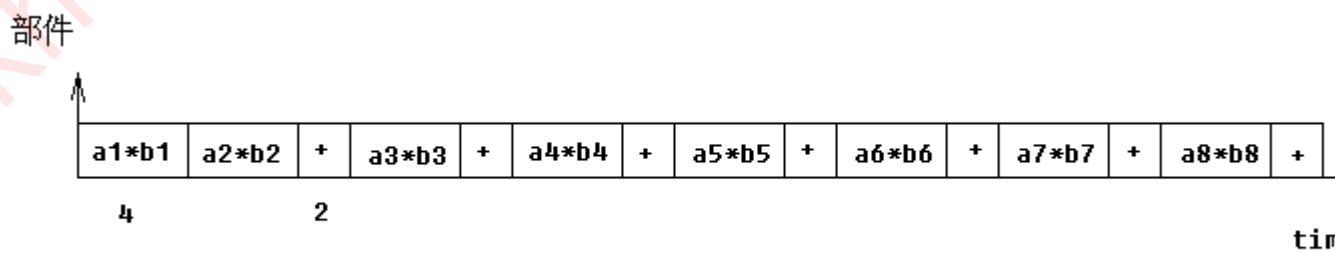
- (1) 通用 PE 的串行 SISD 系统；
- (2) 具有一个加法器和乘法器的多功能并行流水 SISD 系统；
- (3) 有 8 个处理器的 SIMD 系统；
- (4) 有 8 个处理器的 MIMD 系统。

设访存取指和取数的时间可以忽略不计；加与乘分别需要 2 拍和 4 拍；在 SIMD

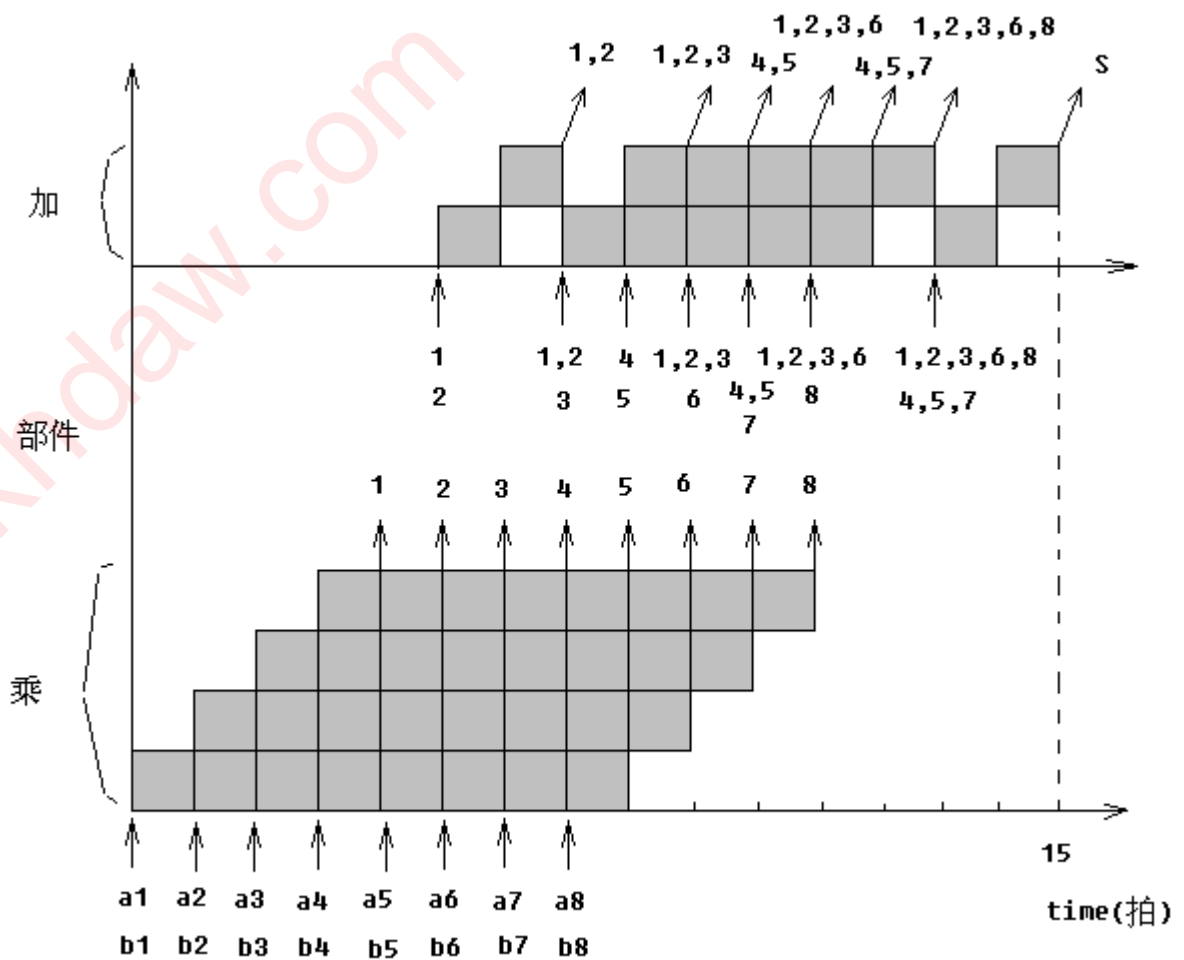
和 MIMD 系统中处理器(机)之间每进行一次数据传送的时间为 1 拍，而在 SISD 的串行或流水系统中都可忽略；在 SIMD 系统中 PE 之间采用线性环形互连拓扑，即每个 PE 与其左右两个相邻的 PE 直接相连，而在 MIMD 中每个 PE 都可以和其它 PE 有直接的通路。

解答：

(1)利用通用 PE 的串行 SISD 系统计算点积所需时间为 46 拍，时空图如下图所示：



(2)利用具有一个加法器和乘法器的多功能并行流水 SISD 系统计算点积所需时间为 15 拍，时空图如下图所示：



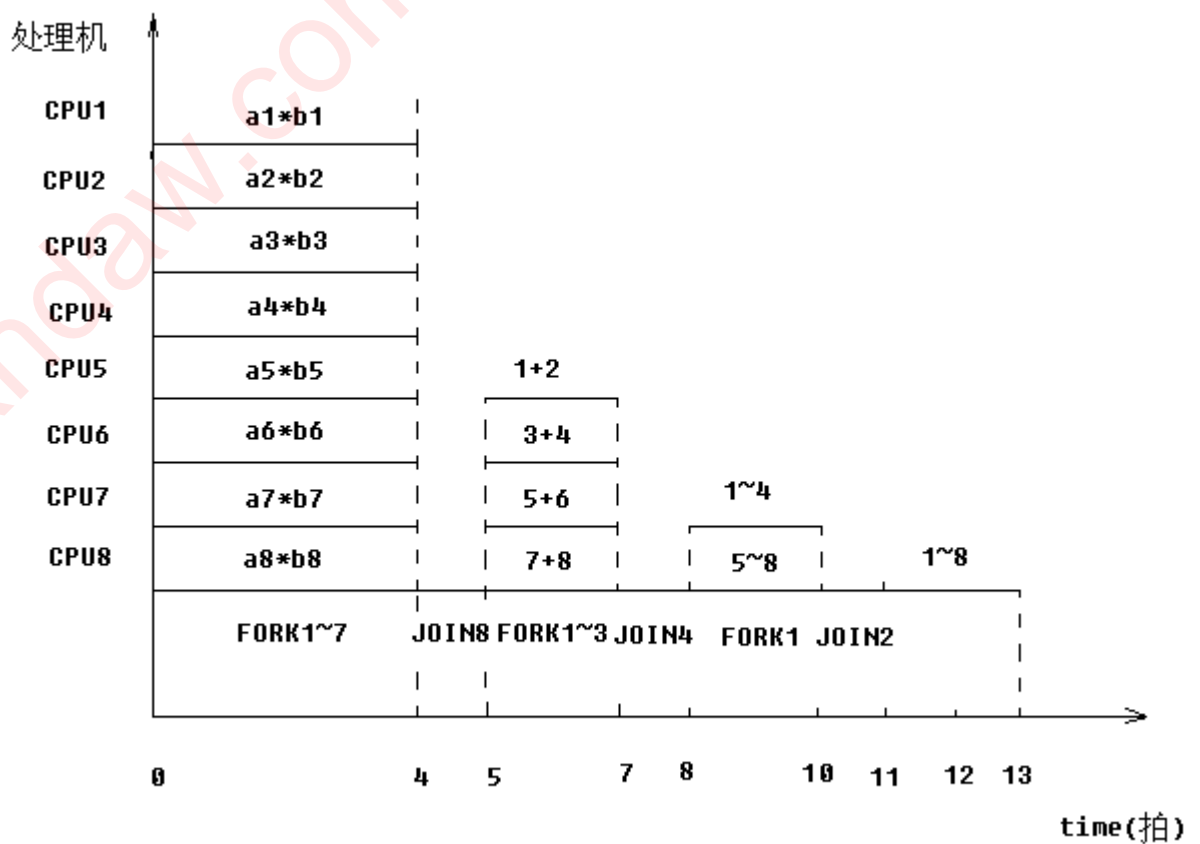
[

(3)利用有 8 个处理器的 SIMD 系统计算点积所需时间为 14 拍, 时空图如下图所示:

upload=gif[uploadimages/200448921199442_as4408a.gif[/upload]

(4)利用有 8 个处理器的 MIMD 系统计算点积所需时间为 14 拍, 时空图如下图所示

示:



9. 设程序有 T 个任务，在 A、B 两台处理机组成的多处理机上运行。每个任务在 A 处理机上执行的时间为 E ，在 B 处理机上执行的时间为 $2E$ ，不考虑机间通讯时间，问如何分配任务，可使系统总执行时间最短？总执行时间最短为多少？

解:

设为 A 处理机分配 I 个任务，为 B 处理机分配 $T-I$ 个任务，则系统总执行时间最短为 $IE=2(T-I)E$ 。解得: $I=2T/3$ 。所以，总执行时间最短为 $2TE/3$ 。

10. 简述多处理机操作系统 3 种不同类型的构形，列出每种构形有优点和缺点以及设计中的问题。

答:

类型	构型	优点	缺点	适用场合
主从型	管理程序只在一台指定的处理上运行	硬件结构简单, 控制简单	对主机的可靠性要求高, 灵活性差	工作负荷固定, 从处理机的能力明显低于主处理机
各自独立型	控制功能分散到多台处理机, 共同完成对整个系统的控制。每台处理机都有一个独立的管理程序(操作系统的内核)在运行。要求管理程序必须是可再入的。	适应分布处理的模块化结构, 对主机依赖性小, 可靠性高, 系统效率较高	尽管每台处理机都有专用表格, 但一些共享表格会增加共享冲突, 加大了开销。实现复杂, 输入输出结构变换需要人为干预, 处理机间负荷的平衡比较困难	松耦合多处理机系统
浮动型	管理程序在处理机间浮动	集中了主从型和各自独立型的优点, 且灵活性高	设计最困难	紧耦合多处理机系统, 同构多处理机系统

第八章课后题

1、简述脉动阵列结构的特点。

答：

- (1) 结构简单，规整，模块化强，可扩充性好；
- (2) 处理单元间数据通信距离短，规则，使数据流和控制流的设计，同步控制均简单规整；
- (3) 脉动阵列机中各处理单元同时运算，并行性极高，可通过流水获得很高的吞吐率；
- (4) 输入数据被多个处理单元重复使用，减轻阵列与外界 I/O 通信量，降低系统对主存和 I/O 系统频宽的要求。
- (5) 脉动阵列结构的构形与特定任务和算法密切相关，具有专用性，限制了应用范围。

2、什么叫控制驱动、数据驱动、需求驱动？

答：

控制流驱动:即指令的执行是在 PC(程序计数器)的控制下,按照事先指定的序列进行的,指令的执行顺序隐含在控制流中。

数据流驱动:即指令的执行是按照数据相关和资源可用性确定的序列进行的,指令的执行基本上是无序的。只要一条指令所需的操作数全部就绪,就可以被激发并执行。

需求驱动:即指令的执行是按照数据需求确定的序列进行的。

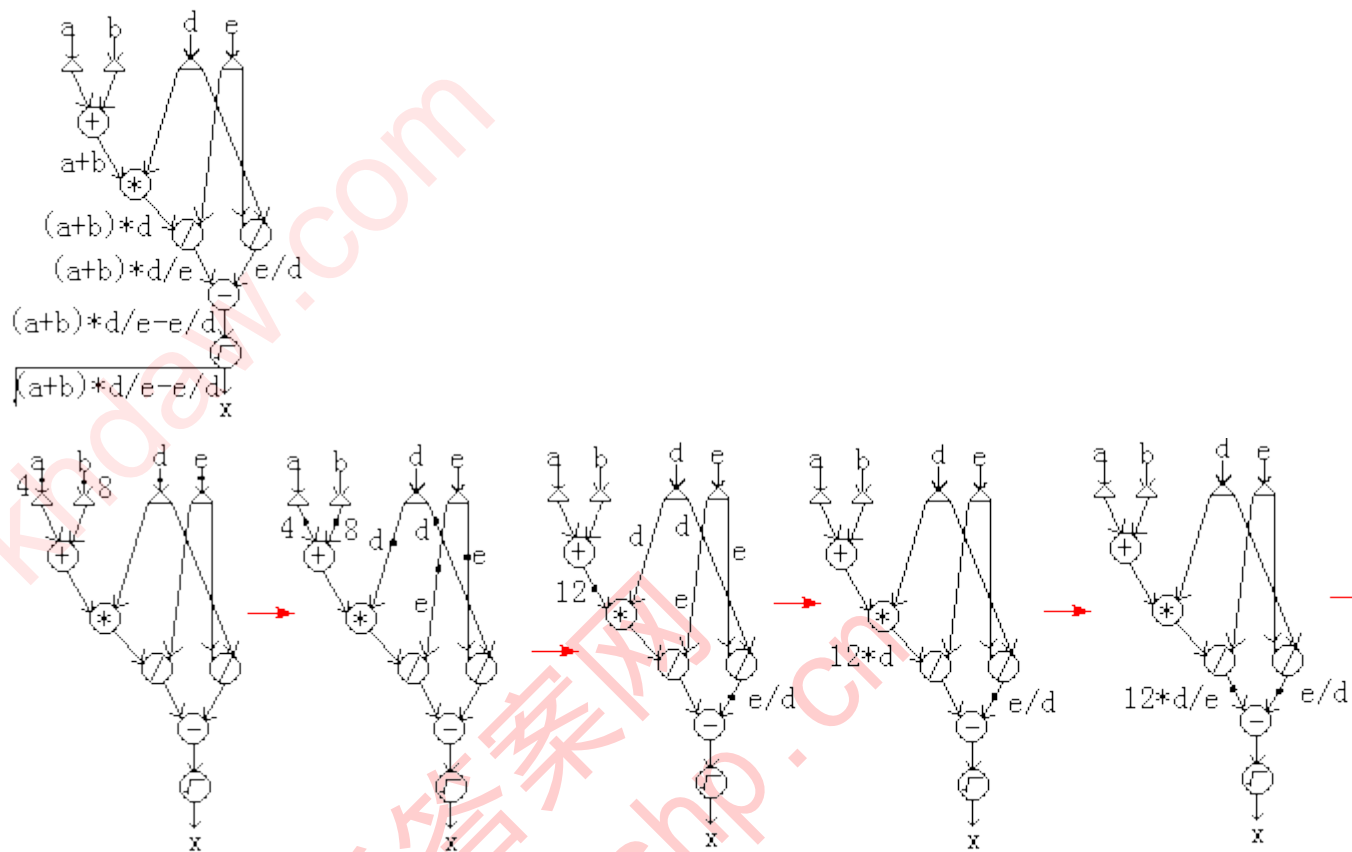
3、什么叫大规模并行处理机 MPP?什么叫机群系统？

答：

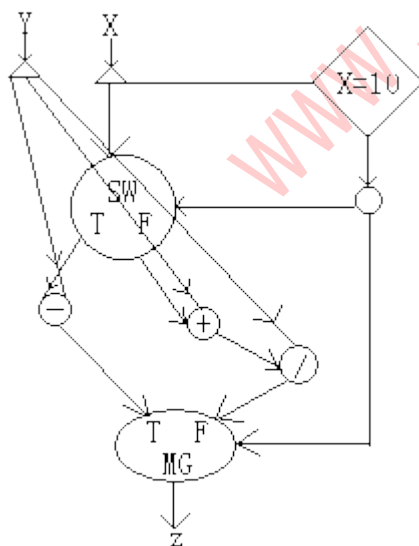
MPP 是大规模并行处理机,指用数百万个高性能,低成本的 RISC 微处理器通过互连网络互连,组成的 SIMD 或 MIMD 系统。

机群系统是将多个高性能工作站或高档微型计算机使用高速通信网络加以互连组成系统。

4、用结构有向图形式画出求解 $x = \sqrt{(a+b) \cdot d / e - e / d}$ 的数据流程图,当 $a=4$ 、 $b=8$ 时,表示出该数据流程图在执行过程。



5、用常用结点画出 $z = \text{IF } X=10 \text{ THEN } X-Y \text{ ELSE } (X+Y)/Y$ 的数据流程图。

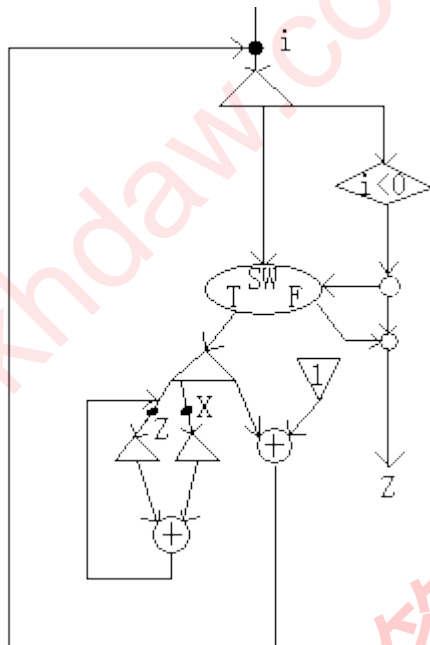


6、画出对应于循环语句
 WHILE $i < 0$ DO
 new $Z := Z + X;$

i:=old i+1

END

迭代结构的数据流程图。



7、静态和动态数据流机的主要区别在哪里？

答：

- (1) 静态数据流机的数据令牌无标号。动态数据流机的数据令牌有标号；
- (2) 静态数据流任意给定时刻当结点操作时每条弧上只能有一个数据令牌、动态数据流机中，任何一条弧上可出现多个不带目标号的数据令牌；
- (3) 静态数据流机中必须设控制令牌以满足要求，动态数据流机中不必设控制令牌，因为令牌有识别时间、先后关系的标号；
- (4) 静态数据流机不支持递归的并发激活，只支持一般循环，动态数据流机支持递归的并发激活；
- (5) 静态数据流机不需硬件完成标记的匹配，动态数据流机需要硬件将标记附加在数据令牌上，并完成对标记的匹配工作。

8.为进行智能信息处理,智能计算机就具有哪些功能,从系统结构上怎样来支持这些功能的实现？

答：

智能机是具有智能的高性能计算机.它是一个知识信息的处理系统.智能机能不断地学习,积累,完善知识,利用知识进行推理,判断和求解问题.它有大容量的知识库,有高度并行处理,多重处理和分布处理能力的多个处理机,是一种结构动态可变,易于扩充的开放式系统,提供有良好的人-机界面和多种智能接口.智能机中 3 个重要的组成部分是知识库机,推理机和智能

接口处理机.

khdaw.com

课后答案网
www.hackshp.cn

khdaw.com